

**UNIVERSITY  
OF OULU**

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

**Perttu Pitkänen**

**AUTOMATIC IMAGE QUALITY ENHANCEMENT  
USING DEEP NEURAL NETWORKS**

Master's Thesis  
Degree Programme in Computer Science and Engineering  
March 2019

**Pitkänen P. (2019) Automatic image quality enhancement using deep neural networks.** University of Oulu, Degree Programme in Computer Science and Engineering. Master's thesis, 66 p.

## **ABSTRACT**

**Photo retouching can significantly improve image quality and it is considered an essential part of photography. Traditionally this task has been completed manually with special image enhancement software. However, recent research utilizing neural networks has been proven to perform better in the automated image enhancement task compared to traditional methods.**

**During the literature review of this thesis, multiple automatic neural-network-based image enhancement methods were studied, and one of these methods was chosen for closer examination and evaluation. The chosen network design has several appealing qualities such as the ability to learn both local and global enhancements, and its simple architecture constructed for efficient computational speed. This research proposes a novel dataset generation method for automated image enhancement research, and tests its usefulness with the chosen network design. This dataset generation method simulates commonly occurring photographic errors, and the original high-quality images can be used as the target data. This dataset design allows studying fixes for individual and combined aberrations. The underlying idea of this design choice is that the network would learn to fix these aberrations while producing aesthetically pleasing and consistent results.**

**The quantitative evaluation proved that the network can learn to counter these errors, and with greater effort, it could also learn to enhance all of these aspects simultaneously. Additionally, the network's capability of learning local and portrait specific enhancement tasks were evaluated. The models can apply the effect successfully, but the results did not gain the same level of accuracy as with global enhancement tasks. According to the completed qualitative survey, the images enhanced by the proposed general enhancement model can successfully enhance the image quality, and it can perform better than some of the state-of-the-art image enhancement methods.**

**Keywords: Photo retouching, color correction, image quality, neural network, affine color transformation, bilateral grid**

Pitkänen P. (2019) Automaattinen kuvanlaadun parantaminen käyttämällä syviä neuroverkkoja. Oulun yliopisto, tietotekniikan tutkinto-ohjelma. Diplomityö, 66 s.

## TIIVISTELMÄ

Manuaalinen valokuvien käsittely voi parantaa kuvanlaatua huomattavasti ja sitä pidetään oleellisena osana valokuvausprosessia. Perinteisesti tätä tehtävää varten on käytetty erityisiä manuaalisesti operoitavia kuvankäsittelyohjelmia. Nykytutkimus on kuitenkin todistanut neuroverkkojen paremmuuden automaattisessa kuvanparannussovelluksissa perinteisiin menetelmiin verrattuna.

Tämän diplomityön kirjallisuuskatsauksessa tutkittiin useita neuroverkkopohjaisia kuvanparannusmenetelmiä, ja yksi näistä valittiin tarkempaa tutkimusta ja arviointia varten. Valitulla verkkomallilla on useita vetoavia ominaisuuksia, kuten paikallisten sekä globaalien kuvanparannusten oppiminen ja sen yksinkertaistettu arkkitehtuuri, joka on rakennettu tehokasta suoritusnopeutta varten. Tämä tutkimus esittää uuden opetusdatan generointimenetelmän automaattisia kuvanparannusmetodeja varten, ja testaa sen soveltuvuutta käyttämällä valittua neuroverkkorakennetta. Tämä opetusdatan generointimenetelmä simuloi usein esiintyviä valokuvauksellisia virheitä, ja alkuperäisiä korkealaatuisia kuvia voi käyttää opetuksen tavoitedatana. Tämän generointitavan avulla voitiin tutkia erillisten valokuvausvirheiden, sekä näiden yhdistelmän korjausta. Tämän menetelmän tarkoitus oli opettaa verkkoa korjaamaan erilaisia virheitä sekä tuottamaan esteettisesti miellyttäviä ja yhtenäisiä tuloksia.

Kvalitatiivinen arviointi todisti, että käytetty neuroverkko kykenee oppimaan erillisiä korjauksia näille virheille. Neuroverkko pystyy oppimaan myös mallin, joka korjaa kaikkia ennalta määrättyjä virheitä samanaikaisesti, mutta alhaisemmalla tarkkuudella. Lisäksi neuroverkon kyvykkyyttä oppia paikallisia muotokuvakohtaisia kuvanparannuksia arvioitiin. Koulutetut mallit pystyvät myös toteuttamaan paikallisen kuvanparannuksen onnistuneesti, mutta nämä mallit eivät yltäneet globaalien parannusten tasolle. Toteutetun kyselytutkimuksen mukaan esitetty yleisen kuvanparannuksen malli pystyy parantamaan kuvanlaatua onnistuneesti, sekä tuottaa parempia tuloksia kuin osa vertailluista kuvanparannustekniikoista.

**Avainsanat:** Kuvien jälkikäsittely, värinkorjaus, kuvanlaatu, neuroverkko, affiini värimuunnos, bilateraallinen ruudukko

# TABLE OF CONTENTS

ABSTRACT

TIIVISTELMÄ

TABLE OF CONTENTS

FOREWORD

LIST OF ABBREVIATIONS AND SYMBOLS

<b>1. INTRODUCTION</b>	<b>7</b>
<b>2. IMAGE ENHANCEMENT</b>	<b>9</b>
2.1. Photographic process . . . . .	9
2.2. Image enhancement methods . . . . .	12
2.2.1. Exposure correction . . . . .	12
2.2.2. Contrast enhancement . . . . .	14
2.2.3. Color saturation enhancement . . . . .	15
2.2.4. Color balance correction . . . . .	16
2.3. Limitations of automatic image enhancement methods . . . . .	17
<b>3. NEURAL NETWORK BASED IMAGE ENHANCEMENT</b>	<b>19</b>
3.1. Neural networks . . . . .	19
3.1.1. Building blocks for neural networks . . . . .	19
3.1.2. Learning process . . . . .	22
3.2. Intelligent image enhancement methods . . . . .	24
3.2.1. Early work . . . . .	24
3.2.2. CNN based image enhancement . . . . .	25
3.2.3. GAN based image enhancement . . . . .	25
3.2.4. Commercial image enhancement software . . . . .	25
3.2.5. Choosing the model for further examination . . . . .	27
<b>4. DEEP BILATERAL LEARNING FOR IMAGE ENHANCEMENT</b>	<b>28</b>
4.1. Affine color transformation . . . . .	28
4.2. Bilateral grid . . . . .	28
4.2.1. Network architecture . . . . .	29
4.2.2. Coefficient prediction . . . . .	30
4.2.3. Full resolution processing . . . . .	31
4.2.4. Applying the enhancement . . . . .	32
4.3. Learning the enhancement . . . . .	33
<b>5. IMPLEMENTATION</b>	<b>34</b>
5.1. Training software . . . . .	34
5.2. Dataset generation . . . . .	34
5.2.1. Existing datasets . . . . .	34

5.2.2. Simulating aberrations . . . . .	35
5.3. Survey software implementation . . . . .	38
<b>6. EVALUATION</b>	<b>39</b>
6.1. Evaluating the learning process . . . . .	39
6.1.1. Learning general enhancements . . . . .	40
6.1.2. Learning portrait specific enhancements . . . . .	44
6.2. Evaluating the aesthetics . . . . .	44
6.2.1. Survey design . . . . .	45
6.2.2. Ranking the methods . . . . .	46
6.2.3. Failure cases . . . . .	48
6.2.4. Success cases . . . . .	48
6.2.5. Survey findings . . . . .	48
6.3. Network complexity's effect on inference time . . . . .	50
<b>7. DISCUSSION</b>	<b>51</b>
<b>8. CONCLUSION</b>	<b>53</b>
<b>9. REFERENCES</b>	<b>55</b>
<b>10. APPENDICES</b>	<b>60</b>

## FOREWORD

The overall goal of this thesis was to edit photographs to produce beautiful results without human input. Clearly beauty cannot be measured, and despite the recent technological advancements, such an abstract concept will probably not be reduced to numbers in the foreseeable future. However, some attempts can be made to mimic photography retouching automatically. After this research, it seems that some of these attempts might even create quite pleasant results.

Neural networks are very promising technology for many fields but seem to be far from perfect when it comes to creating artwork whose value is solely subjective. It is understandable that even the most complex neural networks cannot produce universally appealing results if humans themselves cannot agree on one definition of beauty.

I value the opportunity that Visidon Oy provided for me to work and research this fascinating and challenging subject. It was an educational experience in more aspects than the subject itself. At times the work suffered from a lack of rigor which caused some wasted efforts and dead ends. Nevertheless, the thesis was eventually finished, and I would like to thank everyone involved for their patience. More specifically, I would also like to express my gratitude to my coworkers for their valuable feedback and professional guidance. Special thanks go to the thesis supervisor Janne Heikkilä, technological advisor Jari Hannuksela and the survey participants.

Oulu, March 11th 2019  
Perttu Pitkänen

## LIST OF ABBREVIATIONS AND SYMBOLS

AI	Artificial intelligence
BG	Background
CNN	Convolutional neural network
CPU	Central processing unit
DSLR	Digital single-lens reflex camera
EV	Exposure value
FG	Foreground
GAN	Generative adversarial network
GPU	Graphics processing unit
HDR	High dynamic range
HSV	Color space consisting of channels hue, saturation and value
LDR	Low dynamic range
MSE	Mean squared error
PSNR	Peak signal to noise ratio
RGB	Color space consisting of three channels red, green and blue
ReLU	Rectified linear unit, activation function
UE	Underexposure
$\gamma$	Parameter used in gamma correction to adjust brightness
$\rho(x)$	Transfer function. Sum of 16 scaled ReLU operations
$\sigma(x)$	Activation function determining the output of a neuron
$\tau(x)$	Linear interpolation kernel
$\phi$	Extracted full resolution features
$A_c$	Learned bilateral grid of affine coefficients
$\bar{A}$	Interpolated bilateral grid
$a_{c,i}$	Slopes for transfer function used in guide map creation
$a_{n,m}$	Affine color transformation matrix of size $n \times m$
$b^i$	Biases for a neural network layer
$b'_c$	Additional bias used for guide map creation
$c$	Index for channel in neural network layers
$G_c$	Global features layer
$g$	Guide for bilateral grid
$F_c$	Fused global and local layers
$I$	Input image
$i$	Index for convolutional layer
$L_c$	Local features layer
$L_2$	Loss function calculated as MSE between samples
$M$	Color transformation matrix used for creating guide map
$n_\phi$	Number of extracted features
$n_l$	Number of convolutional layers
$O_c$	Output image
$s$	Stride used in convolution operation
$s_i$	Size of downscaled input for the neural network
$s_b$	Size of the bilateral grid
$t_{c,i}$	Threshold
$w^i$	Weights for a neural network layer or kernel for convolution

# 1. INTRODUCTION

With the increasing capabilities of smartphone cameras, combined with the rising popularity of social media, digital photography has become an easily accessible and commonplace tool for social interaction. Despite the advancements in camera technology, the possibility of user error remains in the photographic process. Often the user may fail to select the aperture, shutter speed or white balance settings in order to produce a good quality photograph. In addition, the limited size of smartphone camera components adds its own challenge to the task of producing high-quality photos.

Image enhancement is the process of applying various visual modifications to images in order to enhance image quality. In essence, this means increasing their resemblance to the real-world scene or make them more visually appealing. This photographic enhancement process, also called post-processing, can be applied to fix these errors. One example of how manual retouching can affect the image quality, is displayed in Figure 1. Despite technological advancements, the enhancement still needs to be done manually by the user. Automation of this process is challenging because image enhancement needs some understanding of the context and semantic information about the photograph. For example, a portrait photograph would require vastly different enhancements as compared to that of a landscape.

One problematic aspect of the automation of image enhancement is that the aesthetic appeal of the photographs is highly subjective. In addition, people have different ideas or preferences about what is a good photograph. This makes it challenging to numerically measure such a concept, which is often needed in order to automate such a process. Alongside the subject and composition of the photograph, the color has a prominent effect on the overall style. The color modifications can change the feel and atmosphere of the photo drastically while being aesthetically pleasing. Some people might find bright and colorful images pleasing, and others might prefer more subdued colors or artistic effects over photorealism.

Typically, photographs are edited with photo retouching software that provides sophisticated tools to process photographs. These tools include features such as color adjustments, image cropping, sharpening and so on. Traditionally these enhancement operations must be applied manually by the user.

In this research the overall goal is to produce high-quality images automatically and computationally efficiently. And the aim is to achieve a natural and pleasing representation of the scene, instead of modifying the image composition, nor applying artistic or stylized effects.

Even though various aspects of automated image quality enhancement have been previously studied, these methods often focus on one specific task such as contrast or color constancy. Commonly the goal of the image enhancement research has been to display the image information as clearly as possible for some technical applications such as medical imaging, where the aesthetic quality of the image is less important than extracting some information from it.

Artificial intelligence, more specifically neural networks, have been applied to carry out tasks that are considered to require human-like perception or extract meaningful information from given data. Artificial intelligence has also been applied successfully to image processing problems in order to enhance image quality, or to automatically generate similar results mimicking different visual styles. The image processing task





Original image

Manually retouched  
version of same image

Figure 1: Color modifications can recover detail and enhance image look. Image source: FiveK dataset [1].

of style transfer is left out of the scope of this thesis, as it is often designed to create new artistic effects such as simulating painting rather than improving the input image quality. Common to the image enhancement methods utilizing neural networks is that they rely on training data which is used as a guidance to produce the desired results. The data must be chosen wisely as it will affect the quality and efficiency of the training procedure.

One specific and popular genre of photography is portrait photography, which can add additional challenges to the image editing process. The color of the face and the surrounding background can have some effect on the human ability to recognize a face from an image or it may cause unwanted meaning or implications for the image. The need for editing the portrait separately from the rest of the image further motivates to investigate the image enhancements capabilities for local enhancements.

The goal of this thesis is to study automated image enhancement towards producing high-quality results. This includes studying the sources of photographic errors as well as both traditional and neural network based methods developed for automated image enhancement. To find out if the chosen neural network based approach can produce appealing results, an extensive evaluation will be conducted.

## 2. IMAGE ENHANCEMENT

Image enhancement or post-processing means digital retouching of the image after it has been created. Often it consists of several transformations that are applied to the image in order to enhance the overall look of the image. It can also be used to fix problems with the photo exposure. Many professional and hobbyist photographers consider post-processing to be an essential part of the photography process. Examples of popular image enhancement software are Adobe Lightroom [2] or RawTherapee [3].

Often the used modifications adjust the colors of the images or enhance other features such as edges or contrast. Some edits might be applied which can change the semantic information of the image, perform inpainting, or introduce new edges. However, the goal of this thesis work is to enhance image quality while preserving the semantic information and structure of the original image.

This chapter takes a closer look into the photographic process and how photographic aberrations may occur. Examples of erroneous photographs are shown in Figure 2. In addition, the multitude of image enhancement methods, their effects, and mathematical representations are studied. This includes the common enhancements that can be found in most image editing software. Additionally, research regarding more sophisticated automated image enhancement is reviewed.

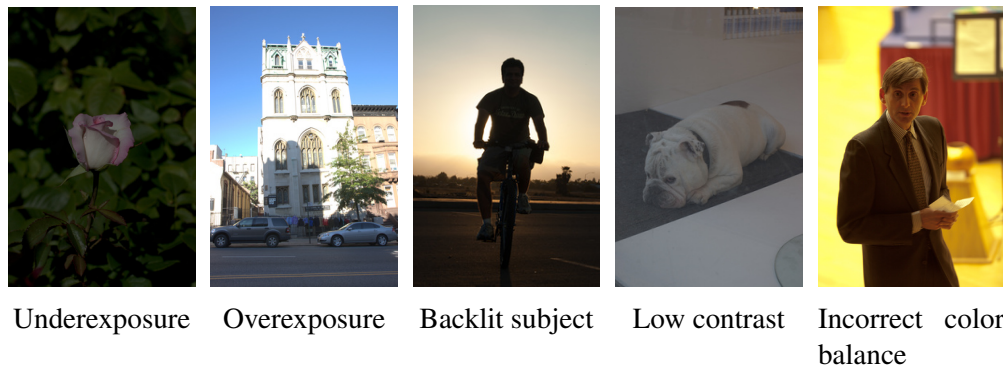


Figure 2: Examples of common photographic flaws. All images are from the FiveK dataset [1].

### 2.1. Photographic process

To understand the need for digital image enhancement it is necessary to understand how the digital photographs are created. Additionally, the possible sources for errors during this process should be noted. Figure 3 shows the process of capturing digital images. This kind of process is typical for conventional digital cameras including mobile phone cameras.

First, the light coming from the scene is captured using camera optics and shutter (1). In this phase, the camera operator or automated camera system is responsible for finding the correct settings for the current scene with the help of a light meter. The exposure can be considered the most critical phase of the photography process, as it determines the amount of light that reaches the sensor creating the image. Several

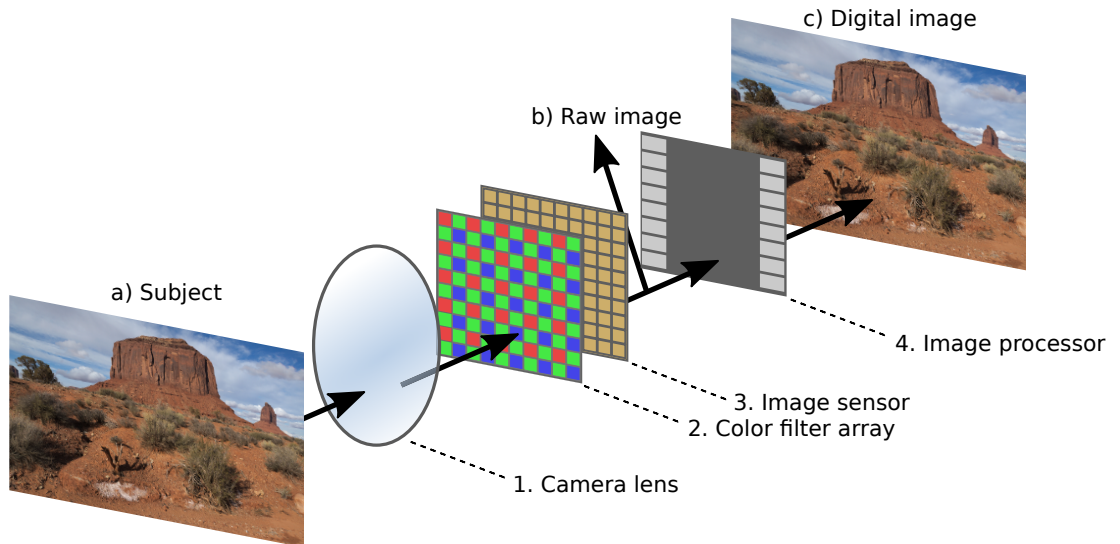


Figure 3: Simplified diagram of digital photography process from a real life scene a) on the left, to a digital image on the right c).

things can go wrong during the photo exposure process: too dark or bright exposure can be caused by an operator error by adjusting the combination of shutter speed, aperture and ISO setting incorrectly. The exposure might also be measured from an irrelevant position in the scene causing the real subject of the photograph to be under- or overexposed.

Technical errors during the exposure can cause photographic flaws with severe information loss or subject displacement. These flaws are not included among the researched problems of this thesis. Such aberrations are, for example, incorrect composition of the subject, unlevel horizon, out of focus image or motion blur caused by a camera or subject movement.

The correct combination of exposure settings depends on the scene's overall lighting and the subject of the scene. A suitable pairing of aperture and shutter speed can be represented using exposure value (EV) as a measure. EV represents numerically the relation of the photographed scene to the shutter speed and aperture as in equation

$$EV = \log_2\left(\frac{a^2}{s}\right), \quad (1)$$

where  $a$  is the relative aperture and  $s$  is the shutter speed. Different scenes or effects require different EVs. Low light situations such as photographs taken at night or indoors with low light require small or negative EV and bright scenes such as in sunny weather require a large EV. [4 p.235]

Figure 4 displays the resulting exposure values from different combinations of aperture and shutter speed. The horizontal lines in the gray grid represents different apertures and the vertical lines correspond to different shutter speeds. Diagonal red lines are the resulting exposure values. The green lines are examples of how the program mode of a digital camera can select the settings. If the camera operator or camera's automatic mode fails to detect the correct exposure value for the scene, the resulting photograph is likely to suffer from incorrect exposure.

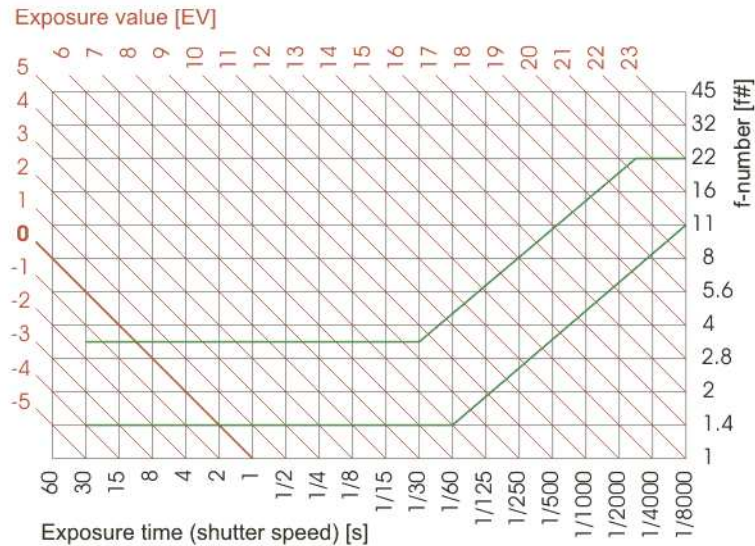


Figure 4: Chart displaying the relation of shutter speed and aperture to the exposure value. Image source: user Darekk2 at Wikimedia Commons<sup>1</sup>.

As a result of the previous steps, an inverted image is created on the sensor plane. Here it is filtered using color filter array (2). The most commonly used color filter is the Bayer filter that contains an array of red, green and blue filters arranged in a similar pattern shown in Figure 3. This filter array allows only certain colored wavelengths of light to pass, thus enabling the creation of color images. [4 p.172]

The image is then captured in digital form using an image sensor (3) lying behind the color filter array. The gain of the image sensor affects the amplification of the signal from the image sensor (called ISO setting). The ISO setting affects the images overall exposure. Lower ISO values require more light than higher values, but high ISO values result in more noisy output. Sensor gain can be changed by the camera user or camera software, which enables more adjustments, but can also cause errors in the image creation process. Furthermore, the quality and size of the camera sensor can limit the image's dynamic range significantly and might result in noisier output than from a larger sensor. The downside of sensor signal amplification is that the noise increases with large ISO values. [4 p.170-171]

This analog signal is then converted into a digital signal for further processing. The resulting latent image from the color array filter data is called raw image (b) which is then passed to the image processor (4). Several image processing functions such as demosaicing, white balance, gamma correction, and noise removal are executed in this phase. Demosaicing algorithm takes the Bayer array data and creates a full-color image from these samples. In the image processing phase, number of things can cause low-quality images. Demosaicing itself has been subject to extensive research and a number of algorithms have been developed to reduce noise or other artifacts. The white balance phase aims to approximate the scene's overall color temperature and adjust the colors accordingly. This can produce noticeable aberration: if the approximated color

<sup>1</sup>[https://en.wikipedia.org/wiki/Exposure\\_value#/media/File:Exposure\\_program\\_chart.gif](https://en.wikipedia.org/wiki/Exposure_value#/media/File:Exposure_program_chart.gif), licensed under CC BY-SA 3.0.

temperature is wrong the image might suffer from an unappealing color cast. [4 p.272-276]

Finally, after the image processing phase, the color image (c) can then be viewed or saved on the device. The latent image can be stored in a large uncompressed raw image format and after processing steps, a compressed file such as jpeg can be created for easier display and archiving purposes. [4 p.1-12]

The limited size of components in a typical smartphone can restrict some aspects of the described image creation process which can result in lower quality images when compared to consumer or professional grade DSLR (digital single-lens reflex) cameras. Usually, smartphones have "slow" lenses compared to conventional cameras, meaning that the amount of light is restricted by the relatively small maximum aperture. In addition, the sensor size of a smartphone camera is only a fraction of a full frame DSLR camera's sensor. This reduces the image resolution and can cause lower quality colors as well as noisier output. [4 p.167]

Most often the photographic flaws are not independent of each other: low exposure images can have low saturation and contrast in addition to the overall darkness of the image. The scene lighting has a major effect on how the photograph is captured: the scene light can be hard or diffused, the subject might be partially or completely in shadow and the color temperature can vary depending on the light source. For examples see Figure 2.

## **2.2. Image enhancement methods**

Color transformations can be applied as a countermeasure for these previously described errors. In this section, more detail is provided on these photographic flaws and previously developed solutions are described alongside some demonstrations of the effects of these enhancements. It should be noted that even though this section is categorized as different operations, these flaws are not separate from each other and the enhancement methods often overlap.

Many of the image enhancements are applied in the RGB color space, which is most often used to represent the pixel color values. In addition, the HSV (Hue, saturation, and value) color space can be used. HSV is designed to represent individual aspects of human color perception instead of separate color channels as in RGB. [4 p.413].

### ***2.2.1. Exposure correction***

As described before, multiple causes can result in incorrect exposure. A specific case of incorrect exposure is the underexposure of a subject with strong background light. Often the photographed scenes can have strong variation in lighting making finding the correct exposure difficult. Especially if the light is measured according to the whole scene instead of the subject. An example case of a backlit subject is shown in Figure 2.

Simple exposure correction is a relatively easy operation. The pixel RGB values can be multiplied with a constant value to produce brighter or darker results. This causes



Figure 5: The under- and overexposed versions of the same scene

a linear modification regarding brightness. This kind of enhancement is demonstrated in Figure 5.

One of the most common nonlinear methods for exposure enhancement is the gamma ( $\gamma$ ) correction which stretches the global intensity of the image [5]. The function of gamma modification is

$$V_{out} = AV_{in}^{\gamma}, \quad (2)$$

where  $V_{in}$  and  $V_{out}$  correspond to the input and output values, and values  $A$  and  $\gamma$  define the shape of the curve. With values of  $\gamma > 1$  the brightness will increase and decrease with  $\gamma < 1$ . A typical shape for a gamma curve used for image brightening is shown in Figure 6.

Some automated methods have been developed to estimate the parameters for gamma correction automatically. Methods have been developed so that the gamma value can be computed either globally or in a local manner.

Yuan and Sun enhance the gamma curve adjustment method in their research [6] by proposing a method that estimates a detail preserving enhancement curve for the image. The method assigns brightness values defined by the zone system [7 p.47-97] to fitting image regions, and estimates the needed parameters for the curve adjustment using this information. The research by FarshbajDoustar and Hassanpour [8] also specializes in locally-adaptive gamma correction. Fitting gamma correction parameters are found by using extracted feature vectors and the K-nearest neighbors algorithm. Later research by Amiri and Hassanpour [9] propose a new approach for gamma correction. They estimate the gamma values for overlapping windows by minimizing the co-occurrence matrices between them. The gamma value that results in the minimum homogeneity is chosen for each window. This operation is applied only in the value-channel of the HSV color model.

Methods based on high dynamic range (HDR) imaging been developed to the purpose of creating well-illuminated images in challenging lighting conditions. The HDR images contain a higher dynamic range representing a greater range of luminance levels than typical devices can display. The HDR images can be tone mapped to produce low dynamic range images with more detail and information than conventional images. [4 p.242-243]

Conventionally multiple exposures with varying shutter speeds i.e. bracketed exposures can be combined into one to produce an HDR image [10]. Hasinoff et al. [11] describe a method which can output a HDR photograph from a burst of photographs. However, this method does not bracket the exposures but align and merge a series of

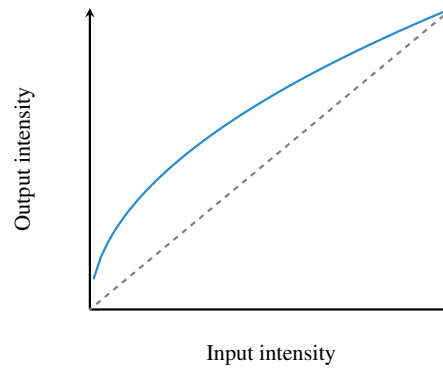


Figure 6: Gamma curve represents the relation between input and output intensities. Dashed line represents no change between input and output.

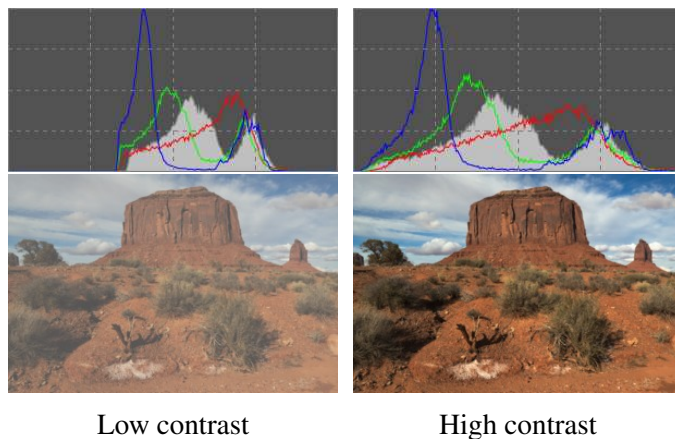


Figure 7: Effect of contrast enhancement and corresponding histogram. Different graph colors represent the corresponding individual RGB-color channel histograms, and the gray histogram represents CIELab luminance histogram. The used image is a0001-jmac\_DSC1459 from the FiveK [1] dataset.

photographs with constant exposure. The method performs tone mapping and denoising for the raw sensor data and merge the burst into a final image.

### 2.2.2. Contrast enhancement

Low contrast occurs commonly alongside too low or too high exposure resulting in an image with a low dynamic range. In the context of image processing contrast means the difference in luminance or color, which increases the human ability to distinguish objects in an image. Among photographers, a rule of thumb for good contrast is that the image histogram should touch both ends of the horizontal axis, representing a large dynamic range.

Traditionally image contrast has been enhanced by the application of histogram equalization. This method stretches out the histogram values selectively in order to distribute the intensities evenly resulting in enhanced contrast [4 p.510]. Histogram enhancement has been developed further including adaptive histogram enhancement [12]

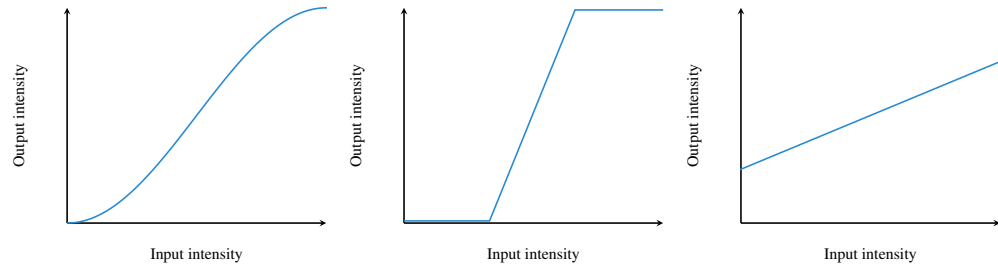


Figure 8: Three different contrast adjustment operations. Left: S-curve contrast enhancement. Middle: White and black point contrast enhancement. Right: White and black points set outside the original dynamic range for contrast reduction.

which aims to reduce the resulting noise that occurs during histogram equalization. A large quantity of the histogram equalization research considers only the enhancement of grayscale images. However, these transformations can also be applied to the color channels separately. The research regarding histogram equalization has also created methods that specialize in color image enhancement such as [13] by Pichon et al, [14] by Bassiou and Kotropoulos and [15] by Han et al.

In addition to the histogram enhancements, S-curve-based enhancement techniques have been developed. These curve based methods have the same principle as with gamma curve in Section 2.2.1. In this case, the curve shape is S-shaped, as the name suggests. Examples of different S-shaped adjustment curves are displayed in Figure 8, and the effect of similar adjustment is demonstrated in Figure 7. Here the histogram is also presented alongside the images.

One simplified way to define S-curve parameters is to set the white and black points for the transfer function. The default white and black levels for no change in output are 0 and 1 respectively. These values can be adjusted to enable contrast related modifications including contrast increase and decrease.

Zavalishin and Bekhtin [16] proposed a local contrast enhancement algorithm which could apply the effects in a manner that emphasizes the most valuable regions in the image. Their method applies the S-curve transformation in an edge aware manner and utilizes a bilateral grid and a saliency map to produce the results.

### 2.2.3. Color saturation enhancement

Saturation is defined as the colorfulness of an area judged in proportion to its own brightness [4 p.78]. Saturation is one of the three components in the HSV color space. In Figure 9 the saturation is enhanced by applying a multiplier value to the saturation component. Saturation value zero would equal a grayscale image.

Usually, the increase of colorfulness can be applied manually within HSV color space. Little research can be found regarding the color saturation enhancement especially in automatic correction. Chiang et al. [17] discuss the problems of modifying the color saturation separately from luminance and suggest an approach that prevents over-saturation. They utilize the YCbCr color model and use it to adjust the luminance and saturation simultaneously.





Figure 9: Effect of saturation adjustment on images.

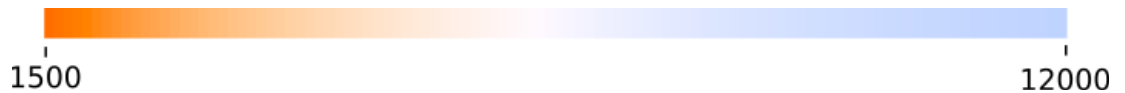


Figure 10: Color temperatures ranging from 1500 K to 12000 K.

It should be noted that the extensively studied task of grayscale image colorization is out of scope for this thesis as it is designed to colorize the images that were originally captured with no color information. This process creates a plausible, but essentially different image, and thus is not included among studied enhancements.

#### 2.2.4. Color balance correction

Varying light sources can result in an incorrect appearance of color in the the resulting photograph. The human eye can adapt to these changes but the same does not apply to the camera sensors. This ability to detect colors in varying illumination conditions is called color constancy. A typical source of colored light is the varying temperature of the light source. The light temperature is defined using the temperature of a black-body radiator which emits light of such color. The values are measured in Kelvin and their range is typically from 1500 K to 12 000 K. This range of color temperatures is visualized in Figure 10. Lower color temperatures correspond to more red colored light and higher color temperatures have light blue color [4 p.44-45].

Color balance or other color cast effects can be manually corrected by adjusting the images overall color temperature with the modification of the RGB values. The target of the color adjustment is to match the illumination of white light. The adjustments can be applied using matrix multiplication in the following manner:

$$\begin{pmatrix} 255/r_w & 0 & 0 & 0 \\ 0 & 255/g_w & 0 & 0 \\ 0 & 0 & 255/b_w & 0 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \\ 1 \end{pmatrix} = \begin{pmatrix} \hat{r} \\ \hat{g} \\ \hat{b} \end{pmatrix}, \quad (3)$$

where  $r_w$ ,  $g_w$  and  $b_w$  are RGB values of a pixel which is believed to be white. Often, this process must be done manually by selecting the point in the scene where pure white or gray is believed to be present. Each color channel of the image is then adjusted proportional to these white point values. Figure 11 demonstrates how color balance adjustments can change the image's appearance. [18]

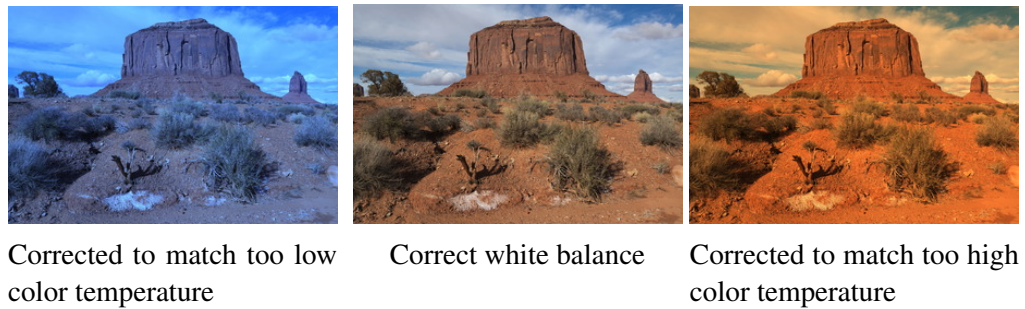


Figure 11: Effects of color balance adjustments.

Many color constancy operations are based on the Retinex theory [19]. Retinex theory states that the human visual system can perceive colors in an image even though the corresponding wavelengths may not be present. This theory states that the human visual system is not based on a simple acquisition of different wavelengths in order to perceive color, but a more complex system where the color relations play an important role. Even when the human visual system is capable of adapting to the illumination conditions, the same does not apply for photographs, creating the need for color adjustments.

Early research based on Retinex theory has developed the so-called Retinex algorithm to estimate the actual underlying colors in the inconveniently lit image. The method is based on the assumption that the maximum intensities at RGB channels correspond to the perfect reflectance. With this assumption, the light source color can be extracted from this white patch. [19]

Color balance correction and color constancy have been subject to extensive research and multiple automated and manual methods have been developed to tackle this problem. Some of these are based on the initial idea of the Retinex algorithm. Along the way, the research such as [20, 21] has developed methods that rely on the physics of reflectance and use this information to estimate the lighting. Furthermore, the gamut-based methods [22, 23] are based on the assumption that the number of observed colors is limited by the illuminant and this information can be used to select the best fitting option from a series of color mappings.

The state-of-the-art color constancy methods rely on learning methodology, meaning that the solution is chosen with the aid of training data. The methods can utilize nearest neighbor search as in [24], regression methods such as kernel regression [25], support vector regression [26] or regression trees [27]. A popular approach is to use neural networks such as described in articles [28, 29, 30]. The survey [31] Kaur and Sharma review several color constancy methods such as described before. Their research suggests that learning based methods such as Color cat method [30] outperform statistic and gamut-based methods.

### 2.3. Limitations of automatic image enhancement methods

Even though the research regarding automatic camera technology has produced many advanced features, the task of manual retouching remains unsolved. Many sophisticated methods have been developed to handle the individual image enhancement tasks,

but most often they are only designed to perform in the specified task and have limited capabilities or require some user input. Even when the methods can enhance image quality, they often cannot be used as a catch-all method to produce high-quality images automatically.

Theoretically, if a system is developed which adds together enhancement methods to cover all of the aberrations that were described, there is no guarantee that such a combination of enhancements will result in a pleasing output. Most importantly, if the method only analyzes the pixel values and image statistics without any information about context or semantics, the algorithm may fail to apply the enhancements in such a way that a human observer finds pleasing. Especially the research regarding color constancy can be seen trending towards applying neural networks to implement intelligent systems where more complex features such as semantic content play an important role. Likewise, this thesis will focus on the image enhancement opportunities that the neural network based methods can provide.

### 3. NEURAL NETWORK BASED IMAGE ENHANCEMENT

The previous chapter described the task and background of image enhancement from a traditional point of view. A notable downside of the previously described methods is that they are limited to some specific approach of analyzing the data, with no semantic understanding of the contents. The current research shows a prominent trend towards the application of neural networks capable of learning the operation. This is especially useful in tasks that require semantic reasoning or "intelligent" processing of the data.

This chapter describes the common underlying algorithms and technologies that are used to build neural networks. In addition, this chapter reviews some state-of-the-art image enhancement solutions that have been developed using these technologies. Typical of these methods is that they aim to implement a system which can mimic a given photographic style or perform predefined enhancement operations. The selected adjustments are chosen according to some understanding of the image contents.

#### 3.1. Neural networks

Artificial neural networks are a subset of machine learning methods. They aim to loosely mimic the neurons in a biological brain. Similar to the biological neurons the input signal is processed in each neuron and the outputs travel from neuron to another. Intertwined combinations of these artificial neurons create the network which would then be trained using some predefined data and eventually produce a presumably desirable output. The output  $y$  of a neuron is

$$y = \sigma(b + \sum_i w_i x_i), \quad (4)$$

where neuron receives  $i$  amount of input values  $x_i$  each of which is multiplied with a corresponding weight  $w_i$ . A bias value  $b$  can be added to shift these values. These weights and biases constitute the adjustable parameters of the neural network. These weighted values are summed together and the resulting value is then passed to an activation function  $\sigma(x)$  which determines the output  $y$  of the neuron, which in turn is one of the inputs for the next neuron in the next layer of the network.

Figure 12 shows the architecture of a simple neural network. Here each of the circles represents a neuron and the network has one fully connected hidden layer. The connections for both layers have weights marked by  $w_{n,n}^i$  and  $w_n^h$  and the hidden layer has biases  $b_1 - b_5$ .

##### 3.1.1. Building blocks for neural networks

With the basic principle of artificial neural networks in mind, several more complex operations have been developed to enhance the learning procedure or to match specific use cases. This section explains several of the most important aspects of neural network design regarding the current task.

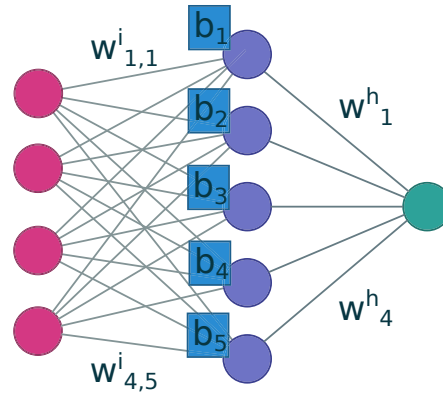


Figure 12: Simple neural network structure.

### Activation function

Activation functions are used to determine the output of a neuron. One commonly used activation function is the rectified linear unit (ReLU), which is used to produce non-linear output. ReLU is defined to be the positive part of its input:

$$\sigma(x) = \max(x, 0), \quad (5)$$

where  $x$  is the input of the activation function [32]. The function returns a value of zero with all input values below zero and returns the input value when it is larger than zero. The plots of ReLU and some other popular activation functions are displayed in Figure 13.

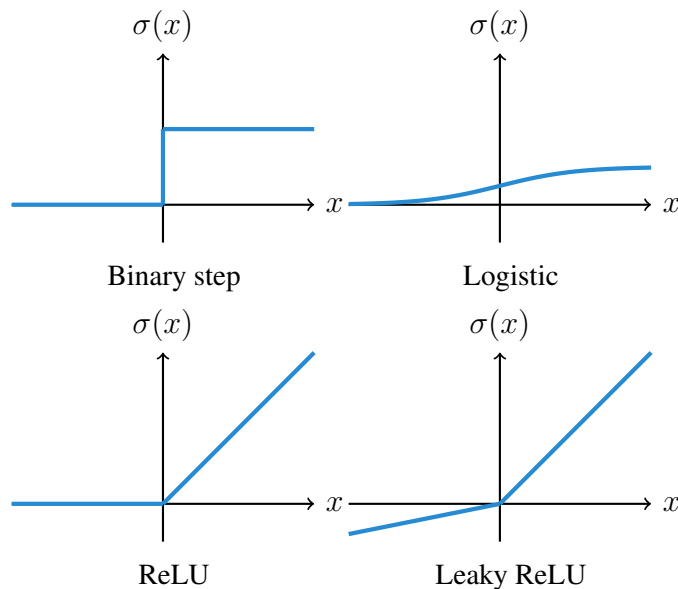


Figure 13: Plots of common activation functions.

### Convolutional neural network

Convolutional neural networks (CNN) are neural networks that have specialized in processing grid-like data such as images. The convolution operations are the back-

bone of this system design. The wide popularity and promising results of the CNNs makes evident the effectiveness of the convolution operation as a part image processing network.

Simply put, convolution is the mathematical operation for two functions producing a third one, describing how one function affects another. In the CNN context the data is usually a 2-dimensional discrete array of image pixels that can be expanded to include the color channels. The convolution operation between the input image  $I$  and the kernel  $w$  has the following definition:

$$(I * w)(x, y) = \sum_{x' y' c'} w_{cc'}[x', y'] I'_c[sx - x', sy - y'], \quad (6)$$

The kernel slides over an image at locations noted by  $x$  and  $y$ . Stride  $s$  determines which image coordinates are iterated over, thus reducing the number of operations. Each value of the input window is multiplied with a value from the kernel indexed by  $x'$  and  $y'$ .  $c$  and  $c'$  index the channel and kernel channel respectively. An example visualization of how convolution operations reduce the spatial dimensions of the 2D array of data is shown in Figure 14. Input array of  $5 \times 5$  size and kernel size  $3 \times 3$  is used. The stride of the convolution operation is 2 and the processing of the array's border values is enabled with padding of size 1.

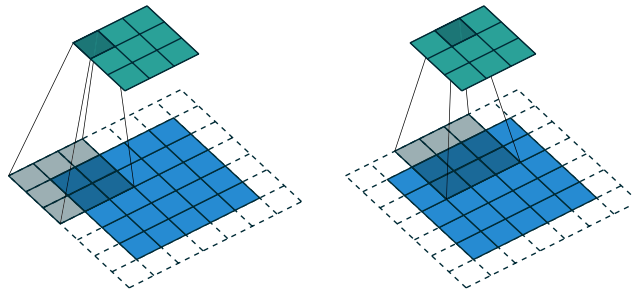


Figure 14: Two steps of convolution operation visualized. Image by Dumoulin and Visin in article [33].

### Pooling

Commonly a pooling layer is placed after the convolution layer. The pooling operation summarizes the values of nearby outputs into one. For example, pooling can report the maximum value, mean or the  $L^2$  norm of the rectangular neighborhood. Average pooling is visualized in Figure 15. The pooling operation reduces the number of parameters in the network thus reducing the amount of computation. Pooling also makes the representation invariant to small changes in the feature locations. [34 p. 335]

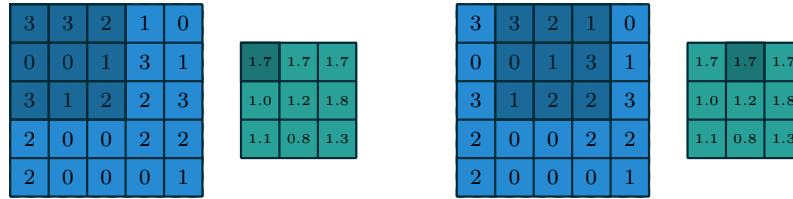


Figure 15: Two steps average pooling. Image by Dumoulin and Visin in article [33].

### Fully connected layer

In a fully connected layer all the neurons in one layer are connected to all neurons in another layer similar as in Figure 12. Traditionally the multilayer perceptron networks consisted of fully connected layers. However, this design can be inefficient with large networks and convolution is most often favored instead of a fully connected layers [34 p. 366]. Usually they can be used as a part of neural network design along with other operations such as convolution.

#### 3.1.2. Learning process

The main advantage of the neural network methodology is its capability of learning arbitrary operations that accomplish the desired output from the given input. To this end, a learning process is carried out to refine the parameters of the network in order to match its output with the training data. The process is done with the use of loss-function, back-propagation, and optimization.

### Loss function

In principle, the loss function is used to measure the performance of the neural network and this information is used to adjust the internal parameters in hopes to produce better results. The loss function (also called the objective function) is the mathematical operation that evaluates the output of the given parameters. For example, the  $L_2$  loss, which is an alternative name for mean squared error (MSE), is commonly used in machine learning applications. MSE is defined as the sum of squared differences between true and predicted values

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_t - y_p)^2, \quad (7)$$

where  $n$  is the number of values.  $y_t$  and  $y_p$  are the reference value and predicted value respectively.

### Back-propagation

As a result of the internal parameters can be evaluated with a loss function, the learning procedure requires a connection between the loss and the network's parameters. How

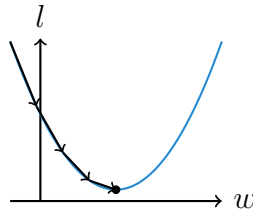


Figure 16: Simple visualization of an optimization problem.

to go back from the loss value through the network and adjust its weights, is done by the application of a process called back-propagation. Back-propagation is basically the method of providing the gradient for the chosen optimization algorithm [34 p. 200].

Many different optimization algorithms have been proposed for the task of finding a sufficient minimum point for the objective function. Examples of such methods are Newton's method or the gradient descent method. [34 p. 80]

Figure 16 visualizes a very simple optimization problem. Here the axis  $w$  represents the one dimensional weights and the axis  $l$  shows the corresponding loss. The minimum point is found by the gradient descend method. Each of the arrows represents one step of the process.

Advanced and more powerful optimization methods have been developed for the purpose of optimizing especially neural networks. Widely used Adam optimizer by Kingma and Ba [35] is a gradient-based stochastic optimization method. The paper describes multiple benefits of the method, such as computational efficiency, invariance to the diagonal rescaling of gradients, and its suitability for a large number of parameters or data.

### Network architecture

There are several different architectural choices for neural network design and the combinations of the different layers, operations and the connections between them. The architectural design choices are highly dependent on the task definition and computational limitations. In addition to the fully convolutional neural networks, examples of image processing network designs are CNN based U-Net [36] or GAN [37] which also can utilize CNN.

U-Net[36] is specific case of convolutional neural network. When typical CNNs have pooling layers the U-Net utilizes upsampling operations and heavy data augmentation by combining different layers of the network.

Generative adversarial network (GAN) can be an especially interesting approach for artistic applications. This framework proposed by Goodfellow et al. consists of generative and discriminative models. The generative model aims to produce data resembling the training data in such a way that the discriminator cannot distinguish the two. [37]

While more complex networks can produce better results, this increase in accuracy comes with a cost. More complexity in the neural network results in slower execution time. He and Sun [38] discuss the trade-offs of different aspects of CNNs with limited execution time. Aspects that affect the time complexity are network depth, the number



of filters/kernels and their size as well as the stride. They characterize the total time complexity of all convolutional layers as

$$O\left(\sum_{i=1}^d n_{i-1} \cdot s_i^2 \cdot n_i \cdot m_i^2\right), \quad (8)$$

where  $i$  indexes the convolutional layer,  $d$  is the depth i.e. the number of convolutional layers,  $n_i$  is the number of filters in the layer and  $n_{i-1}$  is the number of filters in the previous layer. The spatial size of the filter is denoted with  $s_i$  and the feature map size with  $m_i$ .

### 3.2. Intelligent image enhancement methods

With the learning ability of network for performing semantic reasoning and reproducing arbitrary operations, its usefulness in image enhancement task has been confirmed with multiple well-performing methods working as the proof.

This section will briefly describe the work that introduced some important ideas for the current research, and the current state-of-the-art research. Summary of the pros and cons of these methods based on literary review are listed in Table 1.

#### 3.2.1. Early work

Previous research regarding automatic image enhancement has created the basis which was further enhanced with the use of these deep learning neural networks. Even if the methods may have notable limitations they introduce several important ideas and methods which were later enhanced with the use of artificial neural networks.

Kaufman et al. [39] described methods to enhance the image by taking the semantic content into account. This method has several predefined use cases for enhancement including global contrast and saturation correction, face enhancement, sky enhancement, shadowed-saliency enhancement, as well as detail and texture enhancement. Even with intuitive enhancement categories, this method is limited to its hard-coded use cases and style.

Transform recipes [40] is a method which offloads the processing from smartphone to remote server cloud. The system uploads a low-resolution version of the image and its histograms to the cloud where a transform recipe is computed. This recipe is then received in the smartphone where the final effect is applied. This method minimizes the transferred data and can utilize the processing power of a remote server.

Yan et al. [41] claim to have the first image enhancement system using deep neural networks. This method uses discriminative feature descriptors at the pixel, contextual and global levels. The method also includes an algorithm to choose a subset of photos from a large database for high-quality training results.

### 3.2.2. CNN based image enhancement

Convolutional neural networks have proven their efficiency especially in image recognition applications, thus it is only logical that the CNNs are applied to the image enhancement task as well.

Fully convolutional networks are applied in paper [42] where the network is trained with existing filters such as  $L_0$  smoothing, multiscale tone, different photographic style, nonlocal de-hazing, and pencil drawing style. This implementation aims to process the images within the constraints of mobile devices. However, it reached the processing time of 190 ms with 1080p resolution image using a desktop PC.

Ignatov et al. [43] utilized a unique approach to this problem by enhancing smartphone photos by mimicking DSLR quality. They do this with the aid of the generated dataset where the same scene was photographed simultaneously with several smartphones with varying sensor qualities and a full-frame DSLR camera.

In their research Gharbi et al. [44] propose a method that uses input/output pairs of images and aims to predict the affine color transformation which can be applied to the input image. In order to speed up computation time instead of calculating color transformations for each pixel, the method utilizes a bilateral grid in the processing pipeline. In order to apply the final color transformation a coefficient lookup is done from the bilateral grid and applied into the full resolution image.

The specialized case of flash photography for portraits was studied by Capece et al. [45]. In their research, they discuss the problems of using smartphone camera's flash for portrait photography and proposes a CNN which can learn to produce image quality similar to studio quality lighting. The training data used is a dataset consisting of portrait image pairs photographed using smartphone flash and studio lighting.

### 3.2.3. GAN based image enhancement

In the paper by Hu et al. [46] a GAN based approach is applied. Instead of training a neural network with paired input/output pairs the described method uses unpaired data. The main idea is to utilize GAN to generate a sequence of appropriate modifications based on the images current state. The generated image files are evaluated by another network and the generative network's coefficients are adjusted in order to maximize the evaluation score.

Unpaired training with GANs was also applied in paper [47]. This method uses augmented U-Net to generate the results and aims to stabilize the training process of GANs by improving Wasserstein GAN with adaptive weighting scheme.

### 3.2.4. Commercial image enhancement software

Several commercial image enhancement applications are advertised to utilize artificial intelligence for automatic enhancement. Examples of such products are Photolemur [48], Adobe Lightroom [2] and Luminar [49]. However, their design or other technical aspects are not publicly available information.

Table 1: Pros and cons based on literary review of automatic image enhancement methods

Method name	Pros	Cons
Content-aware automatic photo enhancement (Kaufman et al.) [39]	+ Intuitive global and local enhancements	- Very limited use cases - Cannot be trained
Transform recipes (Gharbi et al.) [40]	+ Minimal amount of computing in mobile device	- Requires access to cloud resources
Automatic photo adjustment (Yan et al.) [41]	+ Global and local enhancements + Includes semantic information	- Execution speed not considered
Fast image processing (Chen et al.) [42]	+ Performs well in many applications	- Slow 190 ms processing time for 1080p images with desktop PC
DSLR-quality photos on mobile devices (Ignatov et al.) [43]	+ Specializes in smartphone photography + Improves exposure	- Limited enhancements - Doesn't mimic human retouching
Deep bilateral learning (Gharbi et al.) [44]	+ Can produce multiple effects + Global and local modifications + Low latency processing on mobile device	- Can result in uneven output
Exposure (Hu et al.) [46]	+ No need for paired training data + Separate modifications can be extracted	- Global modifications only - 30 ms processing time on desktop GPU, no mobile tests - Can produce posterization artifacts
Deep photo enhancer (Chen et al.) [47]	+ Good results + No need for paired training data	- Execution speed not considered

Photolemur was designed to perform fully automatic image enhancement with minimal user input. In the software, the user can only specify the strength of the enhancement. The automated enhancements are called color recovery, sky enhancement, exposure compensation, smart de-haze, natural light correction, foliage enhancement, tint perfection, face retouching, JPEG fix, RAW processing, auto lens correction, and auto color temperature. [50]

Additionally, Adobe Lightroom has an automatic adjustment feature. This automatically adjusts the exposure compensation, contrast, highlights, shadows, whites, blacks, clarity and vibrance settings depending on the input image. [2]

The third automatic photo enhancer Luminar also includes AI features to enhance images. According to the Luminar website the Accent-AI-filter can automate traditional controls like shadows, highlights, contrast, tone, saturation, exposure or details. [49]

### ***3.2.5. Choosing the model for further examination***

As described, many solutions have been proposed for the automated image enhancement task. The usage of neural networks is a popular way to enable intelligent processing of the data by learning the enhancement instead of using pre-defined operators. More specifically, CNNs are often applied. Usually, the network's expressiveness requires a complex network consisting of multiple layers. This tends to increase the inference speed of the network. Gharbi et al. [44] discussed proposed a network which is capable fast and edge-aware image enhancement. This proposed computational efficiency seems especially useful for mobile implementations while most other methods did not consider the execution time. For this reason, it was chosen to be the main focus of this research.

## 4. DEEP BILATERAL LEARNING FOR IMAGE ENHANCEMENT

This thesis focuses on testing the capabilities of the method described in paper [44]. It was chosen among the methods described in the previous chapter as based on the literature survey, it suited the initial goal of this research and the results looked promising. The network can process images fast and in a local manner. The basic idea of the system is to learn the color transformation to reproduce the needed modifications from the input image to the desired output image. The color transforms are learned using network consisting of CNN and fully connected layers. It uses a coarse bilateral grid to apply the modifications quickly and in an edge aware manner. This section will cover in detail how the network was built.

### 4.1. Affine color transformation

Article [44] and the research which it is based on utilizes an affine color transformation matrix to apply the needed color modification. Color transforms in RGB color space can be represented with this affine matrix transformation  $a_{n,m}$ . The affine color transformation matrix consists of several coefficients that can be applied to each individual pixel's R, G and B values. This is defined as matrix multiplication

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix} \begin{pmatrix} r \\ g \\ b \\ 1 \end{pmatrix} = \begin{pmatrix} \hat{r} \\ \hat{g} \\ \hat{b} \end{pmatrix}, \quad (9)$$

where a  $3 \times 4$  matrix of affine coefficients  $a_{n,m}$  modify the values of a single pixel's RGB values represented as vector of size  $4 \times 1$ . Appending an additional fourth column  $a_{n,4}$  to the coefficient matrix and fourth-row element 1 to the RGB vector enables the translation operation of the color gamut.

### 4.2. Bilateral grid

The affine transformations which the system will ultimately learn are stored within a bilateral grid. Chen et al. [51] first introduced the bilateral grid data structure to enable edge-aware image processing. The bilateral grid consists of three dimensions width, height, and depth. The width and height dimensions correspond to the 2D position in the image. The depth dimension is called reference range and typically it equals to image intensity. This dimension is also called the guide as it "guides" how the grid values are referenced.

Figure 17 shows an image with four different colors, extracted intensity and corresponding 6 by 4 bilateral grid. X and Y coordinates refer to the width and the height, and Z coordinate is the image intensity. The red dots are samples from the original image projected onto the grid. This structure enables the edge-aware processing of the image, where the modifications are applied on a grid cell basis instead of individual

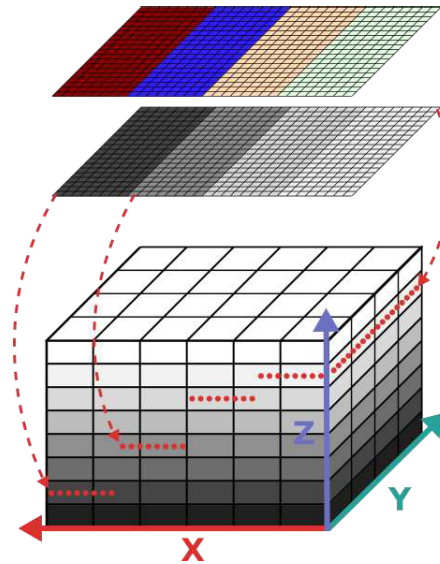


Figure 17: Color image, its intensity and corresponding bilateral grid.

pixels and its neighbors. The pixels within one grid cell can be processed without affecting the values of another grid cell of different intensity even if they were spatially close.

After the operation is applied in the bilateral grid space, the grid is sampled to produce the output image. This operation is called slicing. The slicing samples the affine transformations from grid cells and applies them to the corresponding pixel locations according to the intensity guide i.e. "slices" the grid at intensity dimension. The values applied to each pixel are produced with trilinear interpolation from the coarse grid data in order to yield smooth output. [51]

The bilateral grid data structure has coarser resolution than the image that allows for faster processing than individual pixels, while maintaining information of image edges. The trade-off with the size of the grid cells is that while coarse grid enables faster processing but the output lacks precision. Similar adjustments can also be made for the depth dimension.

Chen et al. further continue their research regarding the bilateral grid in the article [52]. In this research, they use the bilateral grid to approximate several image processing filters. Each cell contains an affine transformation matrix which has been fitted to approximate the image filter's effect. Fitting the suitable values for the affine model is done by solving a linear least squares problem. Finally, the recent research [44] adds the learning neural network structure as a part of the process for finding a suitable bilateral grid containing affine color transformation matrices.

#### 4.2.1. Network architecture

The network architecture was designed to learn color transformations that match the given image input/output pairs. The architecture has two distinct processing paths: low-resolution coefficient prediction to produce the bilateral grid containing suitable color modifications, and a full-resolution path which then apply the modifications to

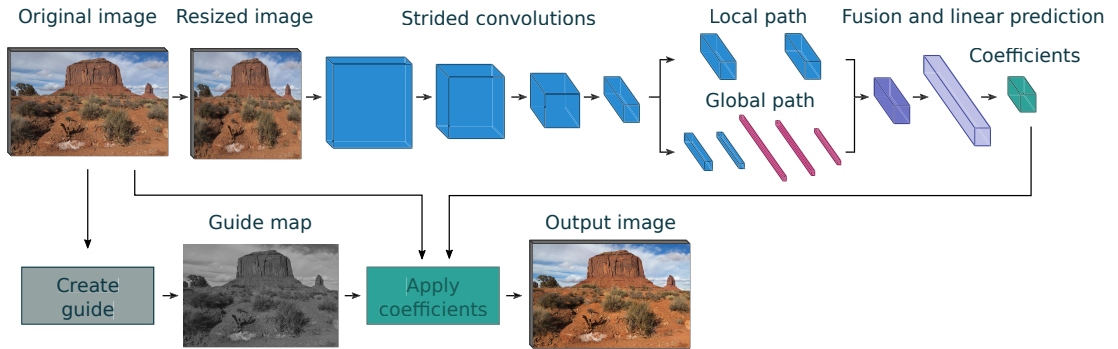


Figure 18: Neural network architecture to learn a bilateral grid of affine coefficients proposed by Gharbi et al. [44].

the image. The network design was developed for fast and efficient processing in mind so the design avoids unnecessary complexity and its spatial dimensions are small compared to the input and output resolution.

#### 4.2.2. Coefficient prediction

To estimate the values for a bilateral grid of affine coefficients using an input image, a convolutional neural network is used. The coefficient prediction path can be seen as the top row of Figure 18. First of all, the 3 color channel image is resized to the predefined size  $s_i$  with bilateral resize operation. The authors used  $s_i = 256 \times 256 \times 3$ . Next, a series of strided convolutions are applied to reduce the size and extract low-level features. The convolutions have stride of 2, use a kernel of size  $3 \times 3$  and use the ReLU activation function. The result  $S^i$  of a convolution operation

$$S_c^i[x, y] = \sigma(b_c^i + \sum_{x', y', c'} w_{cc'}^i[x', y'] S_{c'}^{i-1}[sx + x', sy + y']) \quad (10)$$

uses the preceding layer  $S^{i-1}$  as its input.  $i = 1, \dots, n_l$  is used to index the layers and the number of used layers can be defined with  $n_l$ . The convolution operation is applied in the spatial dimensions  $x$  and  $y$  as well as the channel dimension  $c$ . The  $w^i$  is an array of weights i.e. the kernel. The added biases for the layer are marked by  $b^i$ . The number of needed convolution layers  $n_l$  is

$$n_l = \log_2\left(\frac{s_i}{s_b}\right), \quad (11)$$

and it is defined by the relation of the low-resolution input's size  $s_i$  and the chosen output grid size  $s_b$ . For example, when the dimensions of the layers are reduced from  $256 \times 256 \times 3$  to  $16 \times 16 \times 64$  during the convolution operations (as suggested by the original authors) such operations require four convolutional layers. By modifying the sizes of input and output  $s_b$  the complexity of the network can be adjusted.

In the next phase, the low-level features are processed by splitting the layers into two different paths. This design was inspired by the research done by Iizuka et al. [53] where the split network design is used as a part of an image colorization network.

In this design, the first path is convolutional with a stride of 1 and its purpose is to have spatial information for local features. The path has two convolutional layers. The spatial size and the number of features are kept constant. The second path has convolutional and fully-connected layers designed to learn global features. It has two convolution operations of stride 2 and three fully-connected layers. With the example values used previously the size of the resulting local feature layer  $L_c$  stays  $16 \times 16 \times 64$  and the size of the global feature layer  $G_c$  becomes  $1 \times 64$ .

The 1-dimensional global feature vector and the 3-dimensional array of local features are combined to produce layer  $F_c$  with the operation

$$F_c[x, y] = \sigma(b_c + \sum_{c'} w'_{cc'} G_{c'} + \sum_{c'} w'_{cc'} L_{c'}[x, y]), \quad (12)$$

which takes the last global and local layers  $G_c$  and  $L_c$  and performs a pointwise summation with biases  $b_c$  and weights  $w_{cc'}$  as were in previous operations. This fusion yields a similarly shaped array of features such as after the first convolutional path e.g.  $16 \times 16 \times 64$ . The final learned operation for the coefficient prediction is the  $1 \times 1$  linear prediction

$$A_c[x, y] = b_c + \sum_{c'} F_{c'}[x, y] w_{cc'}, \quad (13)$$

which produces the feature map  $A_c$  that can be interpreted as a bilateral grid of affine coefficients. For example, with the  $16 \times 16 \times 64$  shaped array, the operation produces a final array of coefficients of size  $16 \times 16 \times 96$ . This coefficient array can then be reshaped as an array of shape  $16 \times 16 \times 8$  where each cell contains a 2D affine matrix of size  $3 \times 4$  such as described in Section 4.1, resulting in the final shape of this 5D array to be  $16 \times 16 \times 8 \times 3 \times 4$ .

### 4.2.3. Full resolution processing

In order to apply the predicted coefficients, a guide must be constructed to apply the affine coefficients in an edge aware manner. The full-resolution features  $\phi$  are extracted to create a set of these features with size  $n_\phi$ . The features are extracted from the full-resolution input image in order to produce the guidance map  $g$  and work as regression variables for coefficient prediction. The simplest approach is to use the image color channels as the features.

Guide map values are used for referencing the affine coefficients from the bilateral grid cells. An example of such a guide map is in Figure 19. The input RGB image data can be converted to such a guide. The operation is sometimes called splatting in the literature. This "learned splatting" is done with a simple guide network that works alongside with coefficient prediction layers. The network to produce guide  $g[x, y]$  from image  $I[x, y]$  consists of two operations. First operation

$$g[x, y] = b_c + \sum_{c=0}^2 \rho_c (M_c^\top \cdot \phi_c[x, y] + b'_c) \quad (14)$$



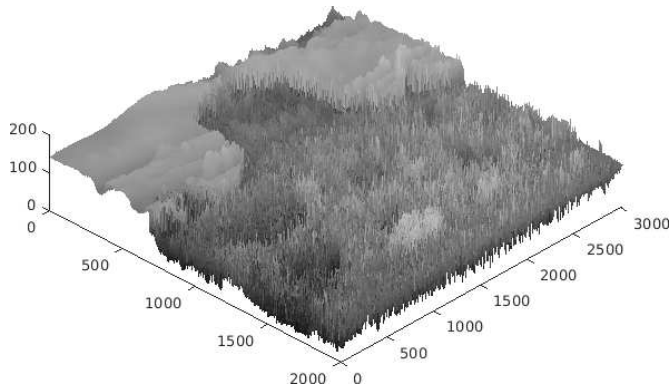


Figure 19: Guide image, where the depth determines how bilateral grid is sampled.

uses a learned color transformation matrix  $M$ , biases  $b$  and  $b'_c$  to construct the guide map. The piecewise linear transfer function  $\rho(x)$  is a sum of 16 scaled ReLU operations:

$$\rho_c(x) = \sum_{i=0}^{15} a_{c,i} \max(x - t_{c,i}, 0), \quad (15)$$

which uses thresholds  $t_{c,i}$  and slopes  $a_{c,i}$ .

#### 4.2.4. Applying the enhancement

By now, all the necessary components, that are needed to the construction of the final enhanced image, have been obtained. These are the input image, a 5D array of affine coefficients and a guide map. The resulting image will be viewed at full resolution so the output quality should also be evaluated at full resolution. The proceeding operation is the bilateral slice, which ultimately determines which of the  $3 \times 4$  affine transformation are applied to individual pixels of the original image.

With coarse spatial dimensions of the bilateral grid such as  $16 \times 16 \times 8$  the values must be interpolated to cover the entire extent of the input image, including the depth dimension. To construct a bilateral grid with the same extent of the original image  $\bar{A}_c[x, y]$  trilinear interpolation is used:

$$\bar{A}_c[x, y] = \sum_{i,j,k} \tau(s_x x - i) \tau(s_y y - j) \tau(d \cdot g[x, y] - k) A_c[i, j, k], \quad (16)$$

Here  $s_x$  and  $s_y$  are the ratios of width and height dimensions of the bilateral grid with respect to that of the original image. Operation  $\tau(\cdot) = \max(1 - |\cdot|, 0)$  is a linear interpolation kernel.  $A_c[i, j, k]$  is the previously inferred bilateral grid. With this operation, each pixel of the input image gets a corresponding affine color transformation. Finally the input image's RGB values are modified with the previously obtained coefficients to produce output image  $O_c$ :

$$O_c[x, y] = \bar{A}_{n_\phi + (n_\phi + 1)c} + \sum_{c'=0}^{n_\phi - 1} \bar{A}_{c' + (n_\phi + 1)c}[x, y] \phi_{c'}[x, y], \quad (17)$$

where the interpolated bilateral grid of affine coefficients  $\bar{A}$  is applied in a channel wise manner to the full-resolution input features  $\phi$ .

### 4.3. Learning the enhancement

At this point in the process, the inferred coefficients were applied and the resulting enhanced image was acquired at full resolution. For the training purpose, a dataset of input/output image pairs is used where the trained operation's effect is present in the target data. To evaluate the output against the training data the  $L_2$  loss is calculated between the input and output images.

The article [44] suggested using the batch size 16 and Adam solver with the learning rate of  $10^{-4}$  while keeping the rest of the parameters the same as was described in the original article of Adam optimization [35].

The training software computes the input/output image difference continuously as a part of the training process. Typically, a separate evaluation set is used to monitor how the network performs with unseen data from the same dataset. The evaluation process is done at one-hour intervals during the training.

## 5. IMPLEMENTATION

To test the capabilities of the method by Gharbi et al. [44] their provided reference software was used for the training procedure and tests. The underlying assumption for learning neural networks is, that the model can only be as good as its training data. For this reason, special effort was put into the dataset generation. This chapter will look into the implementation of the used software and the developed methods to produce training data.

### 5.1. Training software

The publicly available implementation<sup>1</sup> was used for the most part of this research. It is made by the authors and is capable of reproducing the results as proposed in the article.

The program is written in Python programming language and uses Tensorflow [54] machine learning framework. Tensorflow is an open source software library providing computation for multiple platforms such as CPU and GPU and devices like mobile, desktop or server.

To enable parallel GPU processing Tensorflow utilizes the CUDA [55] application programming interface (API). The library is developed by Nvidia Corporation to provide parallel computing using graphical processing units (GPU) to accelerate the computing speed. CUDA works as an interface between the host device and the GPU. The computationally heavy operations use GPU computation kernel functions which are executed parallel in threads.

The implementation<sup>1</sup> has separate CUDA kernel operations to apply the bilateral slice operation as it is applied in the full resolution making it the most computationally heavy individual operation.

### 5.2. Dataset generation

The neural network training procedure is highly dependent on the provided training data. With the idea of learning both global and local transformations in mind, the training datasets should be constructed accordingly. This section describes the use of previously created datasets and the proposed method for creating new datasets for testing both local and global enhancement capabilities of the system.

#### 5.2.1. Existing datasets

For the photograph enhancement purposes, the goal is to mimic human-made modifications. In this case, supervised learning is utilized with a database consisting of input-output pairs of original photographs and manually enhanced versions of them. The most widely used dataset for this purpose is the MIT-Adobe FiveK -dataset [1].

---

<sup>1</sup><https://github.com/mgharbi/hdrnet>



Figure 20: The images enhanced by artists can have distinct styles. All images are from the FiveK dataset [1].

It consists of 5000 photographs and five enhanced versions of each with modifications done by five photography students (called Experts A-E). Examples of these global adjustments are shown in Figure 20. The photographs in the database have a wide range of subjects and often the applied editing styles vary significantly between each expert. Among the Experts A-E, the work of Expert C has been the most popular choice in scientific studies [46, 41] using this dataset. In the same article [1] a user study was conducted where Expert C got the highest ranking. However, the adjustments applied in the dataset are not always consistent, which has also been pointed out by Yan et al. [41]. A distinct style can be seen in Expert C's work, favoring blueish tones.

Another dataset created for learning image enhancements was created by Ignatov et al. as a part of their research [43]. However, their dataset has several limitations: The photographs are taken from slightly different positions of the scene, adding a slight parallax effect between the images. The issue of varying resolutions between devices is resolved by applying matching and cropping to find the corresponding region between the images. An additional limitation is that the quality of the target images is the DSLR camera's default settings with an automatic mode, not the output of a human retoucher.

### 5.2.2. *Simulating aberrations*

The existing datasets have their limitations so an alternative approach is proposed to create suitable training data for learning enhancements. Here the idea is to first collect high-quality photos produced and edited by skilled photographers and use them as the target data. To produce the low-quality "input" data, a randomly selected series of simulated aberrations are applied to the original high-quality images. This will result in pairs of low-quality/high-quality image pairs of the same scene.

The previous datasets were manually created by human retouchers or photographers which limited the size of these datasets. For example, FiveK dataset [1] contains 5000 images. One of the most important advantages of the proposed approach is that it can produce significantly more training data.

### **General de-enhancement dataset**

The proposed idea is to produce a dataset for the task of fixing common photographic errors. The dataset should include samples with each aberration and their combinations as described in Section 2.1. The underlying idea is that the photographs can have multiple faults with varying intensities, but the enhanced or well-exposed images have usually consistent quality. Additionally, the dataset should reflect the fact that not all

photographs have these flaws and so the network would learn to recognize and apply the enhancement in the photographs where the operation is needed.

With this approach, the de-enhancements should be chosen randomly with varying amounts. The applied de-enhancements are low exposure, low contrast, low saturation, and incorrect color balance. Examples of generated images are shown in Figure 21. In addition, the network’s ability to learn the recovery of these faults individually can be tested by generating datasets limited to one de-enhancement at a time.

The original target images were downloaded from publicly available photographs at Flickr [56] photo sharing service. The service is popular among professional and hobbyist photographers, thus providing a good source of high-quality photos to use as a reference for high photographic quality.

A selection of commonly occurring photographic subjects was used as search keywords to construct the dataset. The dataset has a total of 10 613 images downloaded from Flickr consisting of different categories with the following distributions: animals 18.2%, buildings 24.5%, landscapes 20.5%, night 15.7%, still life 21.1%. The categories are chosen to include variety in the dataset and cover many popular photographic subjects.

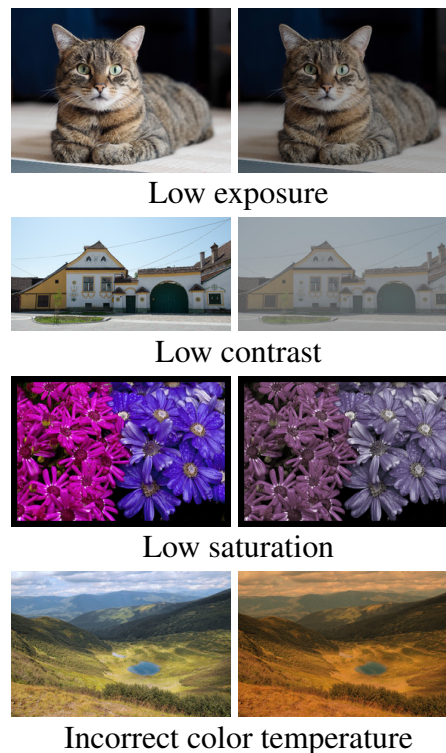


Figure 21: Four samples from the generated de-enhancement datasets. Original image sources: Wolfgang Lonien at Flickr<sup>1</sup>, User Akras at Flickr<sup>2</sup>, User Shebalso at Flickr<sup>3</sup> and Serge Krynysia at Flickr<sup>4</sup>. All modified versions are shared under same license as the original.

<sup>1</sup><https://flic.kr/p/CNYRPL>, licensed under CC BY-NC-SA 2.0.

<sup>2</sup><https://flic.kr/p/KPnne2>, licensed under CC BY 2.0.

<sup>3</sup><https://flic.kr/p/8yBSGC>, licensed under CC BY-SA 2.0.

<sup>4</sup><https://flic.kr/p/nujsJg>, licensed under CC BY-NC-SA 2.0.

### Portrait specific datasets

Supervisory [57] demonstrated their annotation service’s capabilities by providing a free dataset for portrait segmentation [58]. The set has 5711 images of people and the corresponding segmentation data. The original images were gathered from royalty free stock photo service Pexels.

This dataset was used to simulate portrait-specific aberrations for the training data. As the method [44] is capable of local transformations the training dataset should reflect this. The segmentation data was used to simulate special cases for faults that might occur during portrait photography, most importantly the uneven exposure between background and subject. With the aid of segmentation dataset, the exposure of the subject can be modified separately from the background. Figure 22 shows an example of a simulated foreground underexposure.

Further segmentation data was provided by Microwork [59]. This dataset contains several annotated categories: Face, hair, body, and accessories. Similar portrait de-enhancement data can be generated by combining all segmented areas into one portrait area and using it as the foreground mask. With both datasets combined and grayscale images removed the portrait dataset has totally 8947 images and masks.



Figure 22: Sample from the generated underexposed portrait dataset. Image source: User Porapak Apichodilok at Pexels<sup>1</sup>.

While the underexposure of the portrait subject may be common, an additional dataset was constructed where both background and foreground are underexposed with different amounts. Underlying idea being that the network would learn to enhance regions individually depending on whether underexposure is present or not.

With the Microwork [59] face segmentation dataset the face brightening operation can be applied to the original images and produce a dataset for this purpose. The network’s capability for learning a similar dataset was described in the article [44].

### De-enhancement dataset generation software

Many software implementations exist to modify image colors in order to counter such aberrations that may occur. One example of image processing software is the ImageMagick [60] software suite. It includes features such as displaying, converting and editing raster or vector images. A python binding named Wand [61] implements Python programming interface to the ImageMagick features. This enables the easy implementation of Python script which can generate the selected de-enhancements.

<sup>1</sup><https://www.pexels.com/photo/beautiful-blur-camera-capture-348528/>, licensed under the Pexels License.

The low exposure simulation multiplies each of the RGB values with a specified constant below 1 darkening all colors equally. For portrait specific de-enhancement only the mask pixels are multiplied with this value and the same can be applied for the background separately. The darkening coefficient was limited to the range [0.2, 1] where 1 would not have any effect on output. The lower limit was chosen to represent severe underexposure while still being recoverable through exposure adjustment operation.

The color saturation modifications are applied in the HSV color space. Here the saturation component of each pixel is multiplied similarly with a constant value below 1 to simulate the low saturation. In this case, the used coefficients are in the range [0.2, 1] to ensure that the image has at least some amount of color information that can be enhanced.

For the application of different color balance, each pixel value is multiplied with the scaled RGB values representing different color temperatures as defined in Section 2.2.4 ranging between 1500 K and 12000 K.

Simulating low contrast was implemented by randomizing the values of black and white point adjustments. The adjustments to black and white point were implemented symmetrically meaning that the black point decrease and white point increase have the same relative amount e.g. when the black point will be decreased by 50%, the white point will be increased by 50%. The range of values is limited between 0% and 100% so that the transformation curves can produce a lower dynamic range than the original image while retaining information.

### **5.3. Survey software implementation**

For a visual comparison of several images enhanced with various methods, a specific survey software was implemented. In the software, the user is shown two images side by side for quick comparison. The user is then asked to choose which image they prefer of the currently shown pair and several images are shown for all reviewed image comparison tasks. The software was implemented with Python 3.5 using PyQt5 [62] library for implementing the user interface. The survey program randomly picks the images from the image pool and the image display order is also randomized. A screenshot image of the software user interface is presented in Appendix 1.

## 6. EVALUATION

The motivation for this research is to find out how photographs can be enhanced both locally and globally while producing pleasing results. The evaluation process has two approaches: measuring the network’s ability to learn the given operation from a generated input/output image dataset and comparing the output of the proposed model against other state-of-the-art methods.

To monitor how well the network can learn given operation numerical measures can be taken to find out how well the network’s output images match the training target images. It is problematic to determine if the image quality is actually good if no reference target or numerical metric for visual quality is available. The proposed model’s results will be pitted against other research regarding automated image enhancement and will be ranked by human observers using a survey.

Keeping the network parameters equal among compared models is important to ensure a fair evaluation. For the majority of the experiments, the input size is  $512 \times 512$  and the resolution of the output bilateral grid is set to  $32 \times 32$ , while the original article proposed the input resolution of  $256 \times 256$  and output grid size  $16 \times 16$ . The motivation behind this design change is that a larger output grid size of the network should be able to produce finer output than a sparser grid. To keep the number of the network’s convolutional layers the same as proposed by the authors, the downsampled input size is also double of the default value, as defined in the Equation 11. The network’s efficiency is based on its simplicity, so the number of layers is kept small for most of the experiments. For more challenging tasks slightly more complex versions of the network are also used. The different models were trained with the batch size 16 and the training time was limited to approximately 46 epochs for general enhancement tasks and 70 epochs for the more complex portrait specific tasks with a smaller dataset. The general enhancement model which will represent this research in the survey was trained for 77 epochs.

### 6.1. Evaluating the learning process

To test multiple different aspects of low-quality photograph correction, the task has been divided into two main categories: the general enhancements and the portrait specific enhancements. General enhancement means the automatic correction of photographs in similar manner as described in Section 2.2. The network should be able to produce parameters to adjust the level of the enhancement depending on the input subject in order to enhance the quality without introducing uncanny colors or artifacts.

The network was trained using generated datasets as defined in Section 5.2.2 so the learning ability of the network can be monitored for individual aberrations and the combination of them. In this section the task is divided into two categories: general enhancement meaning color transformations that enhance the images overall look globally, and portrait specific enhancement which aims to enhance the portrait area separately from the rest of the image. The portrait specific enhancements focus on fixing the uneven lighting that may occur in portrait photography.

The output quality of the network is evaluated by measuring the PSNR value between the output and the evaluation set and calculating the mean value of the results.



The peak signal-to-noise ratio is a metric that measures the ratio of the maximum power of a signal and corrupting noise in decibels. Typically, PSNR has been used to measure the quality of such cases where there is a clearly defined target image available, such as in the case of image compression. For image quality, PSNR is defined via MSE as in Equation 7. PSNR between the reference image and the noisy image is

$$PSNR = 20\log_{10}(MAX_i) - 10\log_{10}(MSE), \quad (18)$$

where  $MAX_i$  is the maximum possible pixel value e.g. 255 for 8 bit data. The resulting value can be used for quality estimation with higher values meaning less noise. Identical images would result in an infinite PSNR value. PSNR measures the relative noisiness of image quality enhancing the pixel-wise likeness measure of MSE.

Some easily implemented mathematical metric is needed to monitor the training progress, so PSNR is fit for that purpose. In the original article [44] the PSNR values for an acceptable quality level range around 23 - 45 dB for learning different enhancement operations such as filter effects, so similar values can be expected for other enhancement tasks as well.

The PSNR metric is basically one way to denote the MSE value. While the MSE is a popular image comparison metric, it is far from perfect when it comes to evaluating images in a similar manner to a human observer. The article written by Wang and Bovik [63] goes into depth of MSE and its flaws especially regarding perceptual data such as images: The MSE value might be the same with significantly different looking images. Examples of compared images and the PSNR value they produce can be seen labeled "Output" and "Target" in Appendix 2. Due to the limitations of the PSNR, additional qualitative evaluation of the output is needed.

### 6.1.1. Learning general enhancements

The task of correcting photographic errors that affect the images' overall appearance is studied separately from local aberrations. The general enhancement includes editing the image globally with exposure and color related adjustments. The tested models were trained to counter the following aberrations: low exposure, low saturation, low contrast, and incorrect color balance. In addition, the proposed model aims to recognize and fix all of the given aberrations.

With the PSNR as the metric, the network's ability to learn the separate enhancements can be monitored. The use of generated datasets of different de-enhancements allows monitoring of the learning ability of the network regarding different enhancement tasks. The purpose of this process is to find out how well the neural network can learn the tasks of correcting photographic errors individually. The network's output is reviewed using the training data for during each step. The separate training dataset is evaluated at hourly intervals.

The first experiment is to monitor the learning ability of different enhancement tasks individually. For these tests, the specific datasets for individual enhancements were generated by using Flickr and portrait images as the common target data and simulating different aberrations individually producing several different datasets.

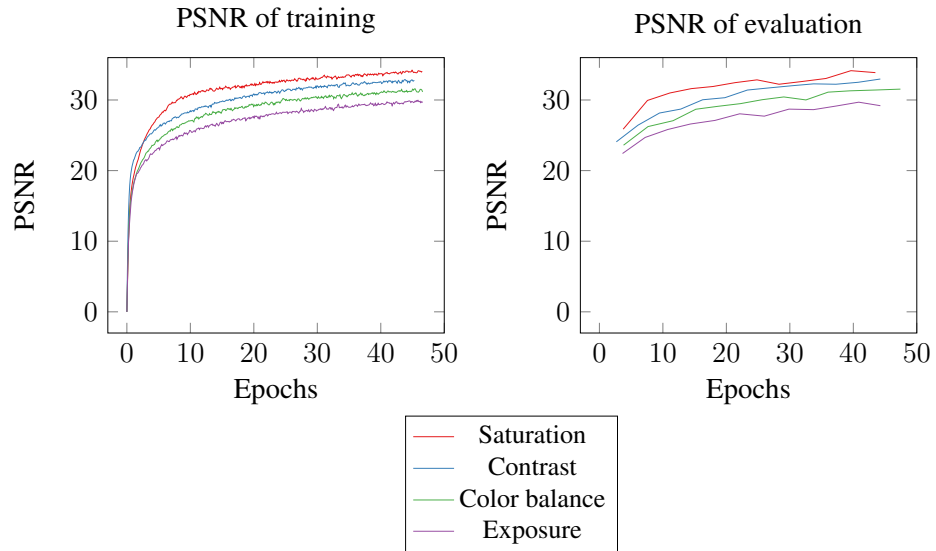


Figure 23: The PSNR values of general enhancement operations monitored using training and evaluation data.

### Learning enhancements individually



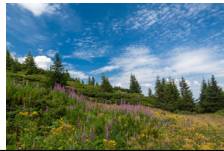
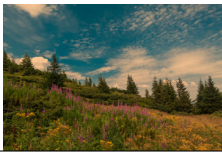

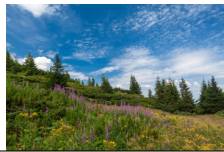


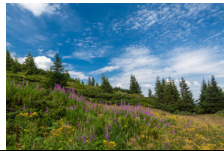


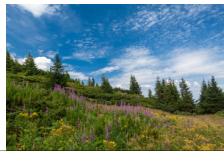
The network performs well in individual global color related enhancement tasks achieving 31.89 dB mean PSNR for evaluation data during 46 epochs. Figure 23 shows the PSNR values during the training process and during recurring evaluation steps. Overall, the easiest individual operation is the saturation increase, which is learned to match the training data fastest. Any particular dataset does not stand out in terms of difficulty so it can be concluded that the network is capable of learning individual enhancement operations from the generated de-enhancement datasets. A visual review of the output images supports this conclusion. In Table 2 the different evaluation set data samples are shown alongside the specific model's output.

### Learning the combined enhancement model

The multiple combined operations are obviously more demanding for the network to learn. With the generated dataset combining de-enhancements with various amounts and using the same complexity as with previous operations (network input size 512 and output grid size 32), the network can reach 24 dB PSNR value during the 46 epochs. This trained model has more variance in the output. While it is generally capable of improving the image quality the output might suffer from some unappealing color casts or lack of saturation.

The combined enhancement model is considered the most useful among the evaluated methods so additional training with alternative parameters is carried out to further improve the results. By decreasing the size of the output grid to  $16 \times 16$  and keeping the size of downscaled input as  $512 \times 512$ , thus increasing the number of convolutional layers from four to five, the network's ability to learn more complex dataset improves. Now the model can reach 28.5 dB accuracy in 46 epochs.

Table 2: The individual trained models can learn to produce consistent quality for different operations. The input images are taken from the generated datasets. Original photo by Sergii Gulenok at Flickr<sup>1</sup>. All modified versions are shared under same license as the original.

Operation	Input	Output	Target	Accuracy
Contrast				28.49 dB
Color balance				27.70 dB
Saturation				25.86 dB
Exposure				24.45 dB

### Learning manual retouches

In addition to the proposed datasets, the network’s ability to learn manually edited photograph dataset was measured. This is a replication of the similar experiment that was done in the article [44]. The chosen dataset was Expert C’s work in the FiveK [1] dataset. The network can learn the distinct style of Expert C more efficiently compared to the proposed dataset. An increase in network complexity has a lower impact on the network’s performance for the Expert C dataset. A comparison of the effects of increased complexity between Expert C, and the proposed dataset is shown in Figure 24. Visual results are included in Appendix 3.

<sup>1</sup><https://flic.kr/p/oZARMV>, licensed under CC BY-NC 2.0.

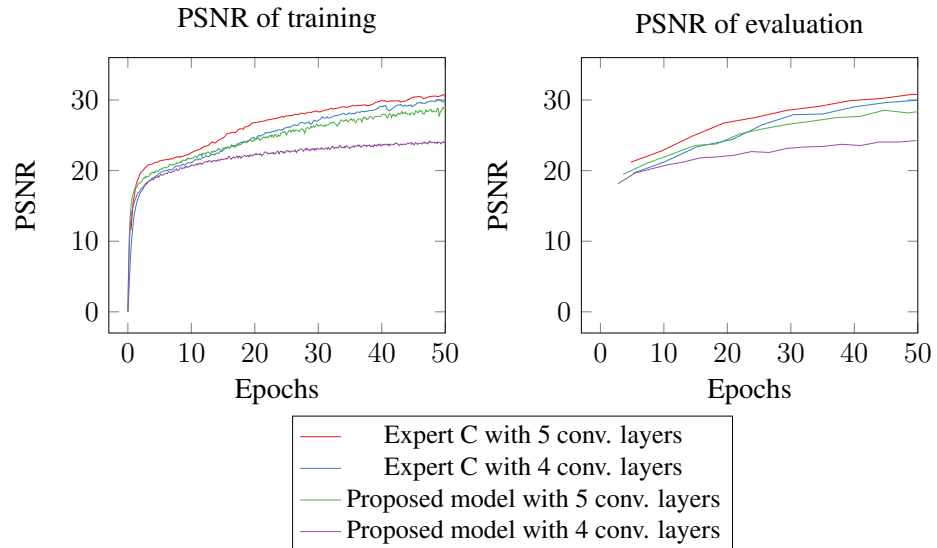


Figure 24: The increase in network complexity has positive effect on its learning ability. Graphs show the PSNR values of general enhancement operations monitored using training and evaluation data.

Table 3: The model for multiple enhancement operations can learn to improve image quality but cannot reach similar level of accuracy as with individual operations especially in the most extreme cases. More images are displayed in Appendix 2. Photo sources Sergii Gulenok at Flickr<sup>1</sup>, Royal A at Pexels<sup>2</sup> and Matt Green at Flickr<sup>3</sup>. All modified versions are shared under same license as the original.

Operation	Input	Output	Target	Accuracy
Proposed model				26.20 dB
				23.80 dB
				17.32 dB

<sup>1</sup><https://flic.kr/p/oZARMV>, licensed under CC BY-NC 2.0.

<sup>2</sup><https://www.pexels.com/photo/close-up-face-fashion-fine-looking-450212/>, licensed under Pexels License.

<sup>3</sup><https://flic.kr/p/bu5TXu>, licensed under CC BY-NC-SA 2.0.

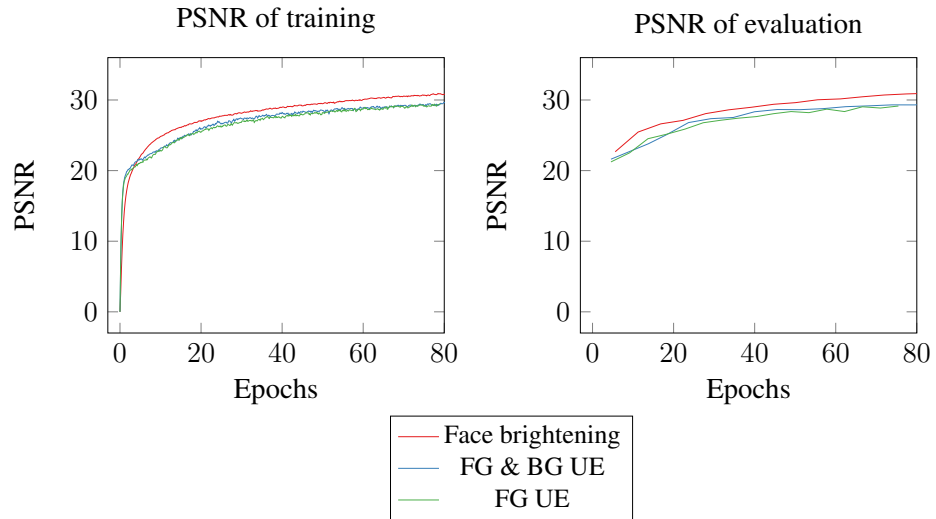


Figure 25: The graphs show PSNR values of portrait specific enhancement operations monitored using training and evaluation data.

### 6.1.2. Learning portrait specific enhancements

For the portrait enhancement task, the evaluated operations are the tasks where exposure is adjusted on portrait and background with varying amounts as well as adjusted in the portrait area only. The generated datasets of foreground and background underexposure (FG & BG UE) as well as the dataset limited to foreground underexposure (FG UE) were used. Furthermore, with the use of the generated face brightening dataset, an experiment similar to the article [44] can be replicated and evaluated.

Quite predictably the local enhancement tasks are learned slower than the global enhancement tasks. While the models trained for global enhancements achieved 31.89 dB mean PSNR during 46 epochs, the portrait specific operation can only achieve 27.94 dB mean PSNR with the same number of epochs. A more detailed overview of the training procedure for different face related enhancement tasks can be seen in Figure 25.

While in the portrait specific cases the learning process is slower and less accurate, the network can successfully learn to reproduce similar effects. The de-enhancements that were applied in the training data were fixed to some extent, and the face brightening experiment can reproduce similar results as were described by the original authors in their article [44]. As for the FG UE and FG & BG UE experiments, the models can improve the image quality, but some dark regions might remain in the output. Examples in Table 4 show how well the results for each task can be reproduced. Additional results are shown in Appendix 4.

## 6.2. Evaluating the aesthetics

Numerical metrics based on image statistics lack any semantic understanding of the image contents, so user surveys shall give more meaningful results. Other learning-based image enhancement methods, such as those described in Section 3.2, can repre-

Table 4: The trained models can learn to enhance images locally towards fixing foreground underexposure as well as combined foreground and background underexposure. In addition, its capability to detect regions such as faces was demonstrated with the face brightening operation. Original photo by user louism60 on Flickr<sup>1</sup>. All modified versions are shared under same license as the original.

Operation	Input	Output	Target	Accuracy
FG UE				22.18 dB
FG & BG UE				18.18 dB
Face brightening				28.37 dB

sent the current state-of-the-art research. Unlike in the evaluation with training data, the images do not have any reference target images, so the results of the enhancement methods are compared with each other.

### 6.2.1. Survey design

To compare and rank several different enhancement methods, a preference survey can be used. In a paired comparison survey, two methods are pitted against each other and combinations spawning all surveyed methods are reviewed. The motivation for a pairwise review was to enable a robust way to review a large number of image pairs quickly.

The forced-choice pairwise comparison produced the most accurate results in comparative research by Mantiuk et al. [64]. In this research, the accuracy of four different image quality assessment methods was measured, and this method resulted in the smallest measurement variance. The overall ranking is determined by the votes each of the compared image set receives. Similar pairwise comparisons were used by Chen et al. to rank their method among other image enhancement methods [47].

The task of choosing a better option between two images should give more meaningful results rather than rating the images one at a time using a numeric scale. If the viewer does not find the subject or the composition of the image pleasant, the reviewer is unlikely to give it a high rating even if the colors had high quality. Randomizing the image display order aims to prevent the users from accustoming to one option during the repetitive task.

<sup>1</sup><https://flic.kr/p/2cvYks1>, licensed under CC BY-SA 2.0.

A random selection of 500 unedited photographs from the FiveK dataset [1] was chosen to be used in the survey. The images of this dataset are suitable for demonstrating enhancement capabilities as no post-processing is applied to them. In addition, this dataset contains the different manually enhanced versions of each of the images. The dataset images are then enhanced with the reviewed methods, and the outputs are compared to each other.

The following enhancement methods are present in the survey:

- Original unmodified images
- Images enhanced using Luminar [49]
- Images enhanced manually by Expert C [1]
- Images enhanced using the Exposure GAN method [46]
- Images enhanced using the proposed model

The reviewed datasets were chosen to represent a variety of enhancement methods. The proposed model is compared to available implementations of state-of-the-art methods as well as manual retouches. The Luminar image enhancement software was chosen to represent the commercial products developed for automated image enhancement. The Exposure [46] method was included to see how a different, GAN based, method compares to the proposed model. For the Exposure method (called Exposure GAN for clarity) the implementation provided by the authors<sup>1</sup> was used with a pre-trained model. The survey also includes manually processed images by the Expert C from the FiveK [1] dataset. The original unedited images are included in the compared images to work as a baseline reference and to detect cases where the method might fail to enhance the image quality. Several examples of the reviewed datasets are displayed in Appendix 5.

Selecting two images at a time for comparison creates 10 different combinations from the given datasets. By sampling 10 images for each combination, each user will see a total of 100 image pairs during the survey process. With this method the users are not burdened with too many images and the random selection of the images can ensure that the reviewed set does not have any bias in the selection.

The user survey is used to ultimately determine how well the model trained with the proposed dataset compares to other state-of-the-art methods in terms of aesthetic quality. In addition, some information can be extracted about which types of image enhancement tasks are performed well for each reviewed method and which fail to produce acceptable quality. This information can be valuable for the future development of the method. To report the results, a preference matrix is constructed to describe how each of the methods compares to each other.

### ***6.2.2. Ranking the methods***

The survey was completed by reviewing 1500 image pairs in total. A larger number of reviewers could have given more statistically robust results, but the overall ranking of

---

<sup>1</sup><https://github.com/yuanming-hu/exposure>

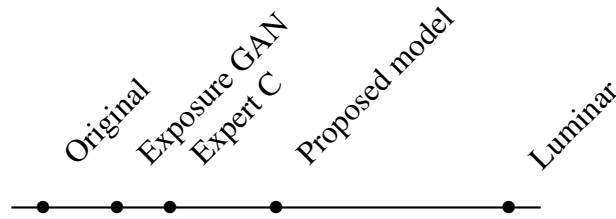


Figure 26: The relative ranking of the methods based on the survey scores.

Table 5: The preference matrix of all reviewed combinations.

	Original	Exposure GAN	Expert C	Proposed model	Luminar
Original	-	87	85	99	131
Exposure GAN	63	-	72	105	120
Expert C	65	78	-	85	104
Proposed model	51	45	65	-	107
Luminar	19	30	46	43	-
Total	198	240	268	332	462

the methods is clear even with this sampling size and any personal stylistic preferences should not have a significant impact on the scoring.

The methods are scored according to the percentage of how often they are preferred over the other methods. The score is generated by scaling the number of times each method "wins" a comparison to the number maximum possible score. There were 150 images reviewed against 4 other enhancements making the theoretical maximum score of 600 (the method wins all of the comparisons).

Clearly, the overall best enhancement method is the Luminar, outperforming all other reviewed methods and it wins the comparison task in 77% of the cases. In second place is the proposed model with 55% score being generally preferred over other reviewed methods except Luminar. The manually enhanced Expert C dataset and the Exposure GAN have lower ranking positions with relative scores of 44% and 40% respectively. The original unmodified image set is predictably at the bottom of the ranking list being preferred over any other modifications in 33% of the cases. Relative ranking of the reviewed methods is displayed in Figure .

The preference matrix is shown in Table 5 displays how the results are ranked, and provide more comprehensive information about how the methods compare to each other. The numbers represent how often the method (column) is preferred over the compared method (row). This preference matrix shows that no enhancement method is perfect, and failures to improve the quality over the original can occur in some cases.

Based on this information, it can be concluded that while the proposed model does not outperform the current state-of-the-art image enhancement in terms of quality, it can improve the image quality well and even its output is preferred to the manually retouched photographs with distinctive stylistic choices. Also, its results are preferred over the Exposure [46] GAN based method as well as the style of Expert C. Special em-



phasis should be put to the fact that Luminar as well as Exposure GAN were designed to work on modern desktop PCs, as the proposed model utilizes network designed for mobile use. Acknowledging its efficient network design, the output of the proposed model has considerably high quality.

### ***6.2.3. Failure cases***

Even if the model trained with the proposed dataset improved image quality in most cases reviewing the specific cases in which the proposed method fails to improve the image quality can provide some valuable insight on how the model or the training dataset can be improved. The simplified overall ranking of these methods gives only a glimpse on how well they actually perform in terms of aesthetic quality.

The preference matrix shown in Table 5 displays how the results are ranked and gives more comprehensive information how the methods compare to each other. The numbers represent how often the method (each column) is preferred over the compared method (each row). This preference matrix shows that no enhancement method is perfect and failures to improve the quality over the original can occur.

For the surveyed image pairs, the proposed model failed to improve image quality in 34% of the cases whereas the Luminar failed only in 12.67% of the comparisons. Expert C and Exposure GAN had worse failure rates of 43.3% and 42% respectively.

In the cases where the proposed method failed to improve image quality, it applied excessive contrast modification or failed to reproduce pleasant colors. Especially, the task of color balance correction seemed to be most often the cause of inconsistencies in perceived image quality. The resulting images might have an unpleasant color cast, or the color temperature was adjusted unnecessarily providing no improvement. In some cases, any significant modification cannot be seen for towards better nor worse quality. Figure 27 shows some examples of such cases where the proposed model was used and the perceived image quality did not improve. These images are presented alongside the assumed reason they failed to impress the survey participant.

### ***6.2.4. Success cases***

While Luminar proved to be the most liked of the reviewed methods, the proposed method was preferred over Luminar in some cases. While the increased contrast was disliked by some of the survey participants, in some cases it could generate better results than Luminar. While the color balance correction proved to be the most challenging operation causing some disliked coloring, it also had some success over Luminar which seems to lack this feature. Examples of some cases where proposed models is preferred over Luminar are shown in Figure 28.

### ***6.2.5. Survey findings***

Some of the participants commented on the difficulty of the task of choosing a preferred image. Picking a better image can be especially difficult if the difference is

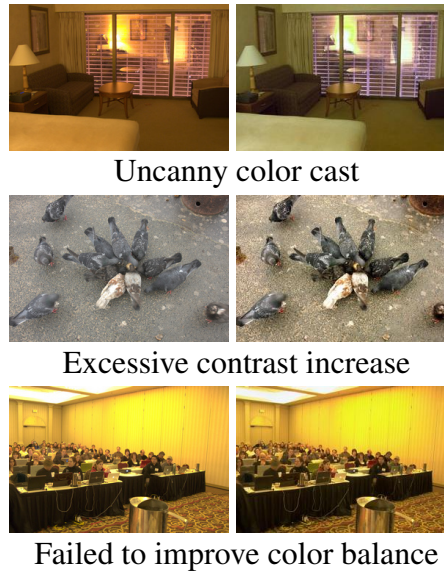


Figure 27: Three examples of failure cases and assumed reasons. Left: unedited image. Right: modified with proposed model. Image sources FiveK dataset [1].



Figure 28: Three examples of cases where the proposed model was chosen over Luminar. Left images: Enhanced with Luminar. Right images: Enhanced with the proposed model. Image sources FiveK dataset [1].

subtle, or if the colors have a notable difference, but neither of the compared images can be considered better.

For the most part, the survey was completed using high quality displays capable of accurate color representation, but the choice of the display was not consistent for all users. This might have added some difficulty to see the subtle differences. For similar surveys the choice of display and the viewing conditions should be kept consistent to avoid any unnecessary deviation in results. Also, some more sophisticated features such as overlaying the images over each other would have been useful during the survey. However, the goal of the survey was to measure the relative quality between images so comparing the quality of two images shown with the same display side by side could still produce valid data for this survey.

The most problematic aspect of automatic image enhancement is the vastly different stylistic preferences people might have. Even when no universal consensus exists about aesthetic quality, some consistency can be seen in the survey results: well exposed images are preferred over underexposed, the color balance correction can improve image quality if done well, increase in contrast and saturation can provide pleasing results if it does not cause loss of detail or uncanny colors. All in all, the survey provided valuable insight on how well each of these methods can perform towards the difficult task of image enhancement, and the ranking of these reviewed methods is clear.

### 6.3. Network complexity's effect on inference time

Section 6.1.1 demonstrated how the increase in network complexity can achieve higher accuracy. However, as discussed in Section 3.1.2 increase in network complexity has a negative effect on the network inference speed. The trade-off between the network complexity and inference time is an interesting research problem especially regarding use cases where the computation time is of the essence.

For this purpose, a test was conducted to measure how increased network complexity affects the inference time. The testing was done using a separate enhancement software. This program was written in C++ language and it utilizes the TensorFlow Lite [65] framework. TensorFlow Lite is a software stack developed especially for mobile or embedded devices, enabling faster computation and smaller model sizes. Previously trained models with different complexities were converted into the TensorFlow Lite format and their inference speed was measured.

Two models with different complexities were tested: one with an input size of  $256 \times 256$  and another with an input size of  $512 \times 512$ . In both cases the resolution is fixed to  $16 \times 16$ . Following the Equation 11, the number of convolutional layers are 4 and 5 respectively. The runtimes were measured using AMD Ryzen 7 CPU with 3.0 GHz clock speed. The average runtime for a model with 4 convolutional layers is 10.11 ms and increasing the number of layers to 5, the average runtime escalates to 27.13 ms.

This rather small test demonstrated how the network's complexity can affect the average inference time. For future research some more comprehensive studies should be conducted, where the image quality and inference time are measured with networks of varying complexities.

## 7. DISCUSSION

During this work special emphasis was given to the training data instead of neural network architecture or implementation. The underlying idea being that even the best network can only be as good as its training dataset. Instead of using an existing dataset such as the FiveK, the proposed method was to simulate well known aberrations and use the vast amount of available high-quality photographs as the training target. The tests and survey proved that the chosen method by Gharbi et al. accompanied by the proposed dataset can work well towards automated image enhancement and outperformed some research of the same field while it fails to outperform the current state-of-the-art image enhancement products here represented by Luminar.

The ill-defined problem of defining aesthetic quality caused most problems during the thesis process. Especially in the early stages of the research, there was not a clear goal of what the research should achieve causing some early trial and error experiments failing to produce meaningful results. However, this challenge gave the inspiration to shift the focus from trying to produce new beautiful enhancements, to finding fixes for known and commonly occurring photographic errors. The idea being behind this design choice is that photographic errors can come in many different forms, but the well-exposed and edited images most often have consistent quality.

This decision proved to be beneficial for this thesis. Previous work regarding automatic image enhancement often concentrates on proving that the methods are capable of mimicking pre-existing image operators or filters with distinct style. The idea of creating a dataset to reflect the multitude erroneous photographs, using the vast amount of publicly available high-quality photography as the target, and implementing the automatic dataset generation method for this purpose can be considered a novel approach. Based on the current literature review of the neural network image enhancement research using similar method has not been extensively tested.

Specific requirements, such as computational efficiency with the possibility of real-time enhancement, added some limitations for the choice of evaluated methods. The work was limited to extensive testing of the chosen method as it was the most promising, providing fast processing and both global and local enhancements. The simplified network design was most likely the biggest limitation on how well it could learn the proposed dataset. Only limited experiments were completed regarding network complexity's effect on the output quality and the network's inference time. The evaluated network was designed with real-time processing in mind, so more comprehensive experiments regarding performance optimizations should be completed in the future work. Regarding the current research, it was most important to find out how well the network can produce appealing results.

During the evaluation phase, the limitations of numerical image quality measures such as PSNR was taken into consideration and it was only used to monitor the network's ability to match the training data. The image ranking survey provided the most valuable insights on how well the network can enhance image quality. Despite the unpredictability of the personal preferences regarding image quality, the ranking was clear even with the selection size of 15 participants. Essentially, the survey could answer to the question of how well the proposed model compares to other research regarding the automated image enhancement. It should be noted that the proposed model and the network was tested against methods that were not designed for fast pro-

cessing with mobile devices. Even with its simple design, the network could perform considerably well.

For future work, the use of specific enhancement operations could be considered to replace the currently used general affine transformation. Instead of solving the bilateral grid of affine transformation matrices, the grid cells could represent vectors containing parameters for different image enhancement operations. For example, operations such as gamma correction and color temperature need on only a few parameters and using the actual operations would simplify the process. Using the bilateral grid to store these parameters would retain the ability for the local and edge aware processing. Using parameters for common enhancement operations instead of a general affine transformation would enable further inspection and adjustment of these values. A similar white box approach was used by the GAN based Exposure method [46]. However, such a design change might make the implementation less efficient in terms of execution time. In addition, the effect of the network's complexity on its computation time could be further evaluated. This is essential if the model is implemented to work in real time on mobile devices.

Much research can be done regarding automated image enhancement, and the unpredictable nature of human preferences will most likely be problematic for future research as well. The work done for this thesis was beneficial in determining that the network can learn the proposed operations, and most importantly a good image quality can be achieved with the used approach.

## 8. CONCLUSION

The original objective of this thesis was to study how well images can be enhanced using automated neural network based enhancement methods. The goal was to produce a similar increase in image quality that would be produced manually by a human retoucher. Studying the current research regarding image processing, including the automatic enhancements, could help understanding why photographs might need re-touching and how it can be achieved.

The method [44] was chosen for its simplified and efficient design and the capability of both global and local enhancements. Its capability of replicating different enhancement operations was evaluated by the original authors, but this thesis focused on testing its ability to apply fixes for common photographic errors.

The network design consists of multiple convolutional and fully connected layers and their combinations and the results are interpreted as a bilateral grid of affine coefficients. This bilateral grid enables local and edge-aware enhancements while being computationally efficient. These design choices were the most appealing features of this network.

While previous research focused on reproducing some pre-existing operations or styles, this thesis focused on the task of producing general and visually pleasing enhancements. With this approach, the proposed dataset generation method was developed and its usefulness in training image enhancement network was put to test. Instead of producing better image quality manually, the proposed method applies randomly chosen de-enhancements to existing high-quality images thus creating a set of high- and low-quality image pairs. The motivation behind this design choice was that these photographic aberrations can be diverse, but the high-quality photographs should have a more consistent quality where the image is bright, and the colors are represented well. The applied de-enhancements were chosen to represent the many aspects of traditional image enhancement. The evaluated operations were exposure correction, contrast adjustment, saturation increase, and white balance correction. Additional testing was done to monitor the network's ability to learn local enhancements. To this end, three portrait specific datasets were generated: two de-enhancement datasets for foreground underexposure, a combined dataset of foreground and background underexposure, and a dataset for the face brightening operation.

During the evaluation, the method was tested in terms of its ability to learn to fix specific pre-defined aberrations and their combination. The models' ability to match the training data was measured in terms of PSNR. In addition, the network's ability of providing fixes for local portrait specific aberrations was tested. For measuring the quality of the general enhancement method, a survey was completed. This worked well towards gaining information on how well the output image quality can work towards producing pleasant results.

The results of the evaluation suggest that the network can learn individual enhancement operations well and local enhancements with slightly lower accuracy. The task of automated and general image enhancement was further evaluated against other state-of-the-art research as well as manual retouches. According to the user survey, a model trained with such a dataset can perform well compared to other research or even manual retouches but cannot reach a similar level as the most sophisticated commercial products. The network complexity's effect on the output quality and inference time were

also discussed, but a more comprehensive study regarding performance optimization is left for future research.

Finally, it can be concluded that the proposed approach can work well towards creating appealing photographs by using the chosen neural network architecture. The unpredictable aesthetic preferences different people have added their own challenges to developing a single method for this task. By focusing on fixing image aberrations, a generally acceptable image quality can be achieved. The survey results suggest that even the most advanced methods fail from time to time and much work can be done regarding automated image enhancement research. Hopefully, the insights and ideas introduced by this thesis can provide a better understanding of the issues surrounding the future research and development of automated image enhancement software.

## 9. REFERENCES

- [1] Bychkovsky V., Paris S., Chan E. & Durand F. (2011) Learning photographic global tonal adjustment with a database of input / output image pairs. In: The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition.
- [2] Adobe Systems Inc. (accessed 11.3.2019.), Buy adobe photoshop lightroom cc | photo editing and organizing softwre. URL: <https://www.adobe.com/products/photoshop-lightroom.html>.
- [3] RawTherapee (accessed 11.3.2019.), Rawtherapee blog. URL: <http://rawtherapee.com/>.
- [4] Allen E. & Sophie T. (2011) The Manual of Photography (Tenth Edition). Focal Press.
- [5] Gonzalez R.C. & Woods R.E. (2006) Digital Image Processing (3rd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [6] Yuan L. & Sun J. (2012) Automatic exposure correction of consumer photographs. In: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato & C. Schmid (eds.) Computer Vision – ECCV 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 771–785.
- [7] Adams A. & Baker R. (1995) The Negative. Bulfinch. URL: [https://books.google.fi/books?id=WUU\\\_ngEACAAJ](https://books.google.fi/books?id=WUU\_ngEACAAJ).
- [8] FarshbafDoustar M. & Hassanpour H. (2010) A locally-adaptive approach for image gamma correction. In: 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), pp. 73–76.
- [9] Amiri S.A. & Hassanpour H. (2012) Article: A preprocessing approach for image analysis using gamma correction. International Journal of Computer Applications 38, pp. 38–46. Full text available.
- [10] Mertens T., Kautz J. & Van Reeth F. Exposure fusion: A simple and practical alternative to high dynamic range photography. Computer Graphics Forum 28, pp. 161–171. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2008.01171.x>.
- [11] Hasinoff S.W., Sharlet D., Geiss R., Adams A., Barron J.T., Kainz F., Chen J. & Levoy M. (2016) Burst photography for high dynamic range and low-light imaging on mobile cameras. ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 35.
- [12] Pizer S.M., Amburn E.P., Austin J.D., Cromartie R., Geselowitz A., Greer T., ter Haar Romeny B., Zimmerman J.B. & Zuiderveld K. (1987) Adaptive histogram equalization and its variations. Computer Vision, Graphics, and Image Processing 39, pp. 355 – 368. URL: <http://www.sciencedirect.com/science/article/pii/S0734189X8780186X>.



- [13] Pichon E., Niethammer M. & Sapiro G. (2003) Color histogram equalization through mesh deformation. In: Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429), vol. 2, vol. 2, pp. II–117.
- [14] Bassiou N. & Kotropoulos C. (2007) Color image histogram equalization by absolute discounting back-off. *Computer Vision and Image Understanding* 107, pp. 108–122.
- [15] Han J., Yang S. & Lee B. (2011) A novel 3-d color histogram equalization method with uniform 1-d gray scale histogram. *IEEE Transactions on Image Processing* 20, pp. 506–512.
- [16] Zavalishin S.S. & Bekhtin Y.S. (2018) Visually aesthetic image contrast enhancement. In: 2018 7th Mediterranean Conference on Embedded Computing (MECO), pp. 1–4.
- [17] Chiang J.S., Hsia C.H., Peng H.W. & Lien C.H. (2014) Color image enhancement with saturation adjustment method. *Journal of Applied Science and Engineering* 17, pp. 341–352.
- [18] Viggiano J.A.S. (2004), Comparison of the accuracy of different white-balancing options as quantified by their color constancy. URL: <https://doi.org/10.1117/12.524922>.
- [19] Land E.H. (1978) The retinex theory of color vision. *Scientific American* 237, pp. 108–28.
- [20] Tan R., Nishino K. & Ikeuchi K. (2004) Color constancy through inverse-intensity chromaticity space. *Journal of the Optical Society of America. A, Optics, image science, and vision* 21, pp. 321–34.
- [21] Finlayson G.D. & Schaefer G. (2001) Solving for colour constancy using a constrained dichromatic reflection model. *International Journal of Computer Vision* 42, pp. 127–144.
- [22] A. Forsyth D. (1990) A novel algorithm for color constancy. *International Journal of Computer Vision* 5, pp. 5–35.
- [23] Finlayson G. & Hordley S. (2000) Improving gamut mapping color constancy. *IEEE Transactions on Image Processing* 9, pp. 1774–1783.
- [24] Joze H.R.V. & Drew M.S. (2014) Exemplar-based color constancy and multiple illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, pp. 860–873.
- [25] Agarwal V., Gribok A.V., Koschan A. & Abidi M.A. (2006) Estimating illumination chromaticity via kernel regression. In: 2006 International Conference on Image Processing, pp. 981–984.
- [26] Funt B. (2006) Estimating illumination chromaticity via support vector regression. *Journal of Imaging Science and Technology* 50, p. 341.

- [27] Cheng D., Price B., Cohen S. & Brown M.S. (2015) Effective learning-based illuminant estimation using simple features. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1000–1008.
- [28] Funt B.V., Cardei V.C. & Barnard K. (1996) Learning color constancy. In: Color Imaging Conference.
- [29] Barron J.T. (2015) Convolutional color constancy. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV '15, IEEE Computer Society, Washington, DC, USA, pp. 379–387. URL: <http://dx.doi.org/10.1109/ICCV.2015.51>.
- [30] Banić N. & Lončarić S. (2015) Color cat: Remembering colors for illumination estimation. *IEEE Signal Processing Letters* 22, pp. 651–655.
- [31] Kaur H. & Sharma S. (2016) A comparative review of various illumination estimation based color constancy techniques. In: 2016 International Conference on Communication and Signal Processing (ICCSP), pp. 0486–0490.
- [32] Nair V. & Hinton G.E. (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omnipress, USA, pp. 807–814. URL: <http://dl.acm.org/citation.cfm?id=3104322.3104425>.
- [33] Dumoulin V. & Visin F. (2016) A guide to convolution arithmetic for deep learning. *ArXiv e-prints* .
- [34] Goodfellow I., Bengio Y. & Courville A. (2016) *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- [35] Kingma D.P. & Ba J. (2014) Adam: A method for stochastic optimization. *CoRR abs/1412.6980*. URL: <http://arxiv.org/abs/1412.6980>.
- [36] Ronneberger O., Fischer P. & Brox T. (2015) U-net: Convolutional networks for biomedical image segmentation. *CoRR abs/1505.04597*. URL: <http://arxiv.org/abs/1505.04597>.
- [37] Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A. & Bengio Y. (2014) Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, Cambridge, MA, USA, pp. 2672–2680. URL: <http://dl.acm.org/citation.cfm?id=2969033.2969125>.
- [38] He K. & Sun J. (2014) Convolutional neural networks at constrained time cost. *CoRR abs/1412.1710*. URL: <http://arxiv.org/abs/1412.1710>.
- [39] Kaufman L., Lischinski D. & Werman M. (2012) Content-aware automatic photo enhancement. *Comput. Graph. Forum* 31, pp. 2528–2540. URL: <http://dx.doi.org/10.1111/j.1467-8659.2012.03225.x>.

- [40] Gharbi M., Shih Y., Chaurasia G., Ragan-Kelley J., Paris S. & Durand F. (2015) Transform recipes for efficient cloud photo enhancement. *ACM Trans. Graph.* 34, pp. 228:1–228:12. URL: <http://doi.acm.org/10.1145/2816795.2818127>.
- [41] Yan Z., Zhang H., Wang B., Paris S. & Yu Y. (2014) Automatic photo adjustment using deep neural networks. CoRR abs/1412.7725. URL: <http://arxiv.org/abs/1412.7725>.
- [42] Chen Q., Xu J. & Koltun V. (2017) Fast image processing with fully-convolutional networks. CoRR abs/1709.00643. URL: <http://arxiv.org/abs/1709.00643>.
- [43] Ignatov A., Kobyshev N., Vanhoey K., Timofte R. & Gool L.V. (2017) Dslr-quality photos on mobile devices with deep convolutional networks. CoRR abs/1704.02470. URL: <http://arxiv.org/abs/1704.02470>.
- [44] Gharbi M., Chen J., Barron J.T., Hasinoff S.W. & Durand F. (2017) Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)* 36, p. 118.
- [45] Capece N., Banterle F., Cignoni P., Ganovelli F., Scopigno R. & Erra U. (2019) Deepflash: Turning a flash selfie into a studio portrait. CoRR abs/1901.04252. URL: <http://arxiv.org/abs/1901.04252>.
- [46] Hu Y., He H., Xu C., Wang B. & Lin S. (2017) Exposure: A white-box photo post-processing framework. CoRR abs/1709.09602. URL: <http://arxiv.org/abs/1709.09602>.
- [47] Chen Y.S., Wang Y.C., Kao M.H. & Chuang Y.Y. (2018) Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, pp. 6306–6314.
- [48] Photolemur (accessed 11.3.2019.), Photo enhancement | automatic photo enhancement software | image enhancing. URL: <https://photolemur.com>.
- [49] Skylum LLC (accessed 11.3.2019.), Luminar 2018 - the best photo editing software for mac & pc | skylum. URL: <https://skylum.com/luminar>.
- [50] Photolemur (accessed 11.3.2019.), How photolemur works | photo enhancement. URL: <https://photolemur.com/technology>.
- [51] Chen J., Paris S. & Durand F. (2007) Real-time edge-aware image processing with the bilateral grid. In: *ACM SIGGRAPH 2007 Papers, SIGGRAPH '07*, ACM, New York, NY, USA. URL: <http://doi.acm.org/10.1145/1275808.1276506>.
- [52] Chen J., Adams A., Wadhwa N. & Hasinoff S.W. (2016) Bilateral guided up-sampling. *ACM Trans. Graph.* 35, pp. 203:1–203:8. URL: <http://doi.acm.org/10.1145/2980179.2982423>.

- [53] Iizuka S., Simo-Serra E. & Ishikawa H. (2016) Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016)* 35, pp. 110:1–110:11.
- [54] Google Inc. (accessed 22.1.2019.), Tensorflow. URL: <https://www.tensorflow.org>.
- [55] NVIDIA Corporation (accessed 11.3.2019.), Cuda zone. URL: <https://developer.nvidia.com/cuda-zone>.
- [56] SmugMug Inc. (accessed 11.3.2019.), Home | flickr. URL: <https://www.flickr.com/>.
- [57] Deep Systems LLC (accessed 11.3.2019.), Supervisely - web platform for computer vision. annotation, training and deploy. URL: <https://supervisely.com/>.
- [58] Supervise.ly, Hacker Noon (accessed 11.9.2019.), Releasing "supervisely person" dataset for teaching machines to segment humans. URL: <https://hackernoon.com/releasing-supervisely-person-dataset-for-teaching-machines-to-segment-humans-1f1fc1f28469>.
- [59] General Blockchain Inc. (accessed 11.3.2019.), Computer vision annotation services for computer vision. URL: <https://microwork.io/>.
- [60] ImageMagick Studio LLC (accessed 11.3.2019.), Convert, edit, or compose bitmap images @ imagemagick. URL: <http://www.imagemagick.org/>.
- [61] Minhee H. (accessed 11.3.2019.), Wand - wand 0.5.0. URL: <http://docs.wand-py.org/en/0.5.0/>.
- [62] Riverbank Computing Ltd. (accessed 11.3.2019.), Riverbank | software | pyqt | what is pyqt? URL: <https://riverbankcomputing.com/software/pyqt/intro>.
- [63] Wang Z. & Bovik A.C. (2009) Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine* 26, pp. 98–117.
- [64] Mantiuk R.K., Tomaszewska A. & Mantiuk R. (2012) Comparison of four subjective methods for image quality assessment. *Comput. Graph. Forum* 31, pp. 2478–2491. URL: <http://dx.doi.org/10.1111/j.1467-8659.2012.03188.x>.
- [65] Google Inc. (accessed 11.3.2019.), Tensorflow lite | tensorflow. URL: <https://www.tensorflow.org/lite/>.

## 10. APPENDICES

- Appendix 1. Screenshot image of the survey user interface
- Appendix 2. Additional visual result of the proposed enhancement mode
- Appendix 3. Additional visual result of the Expert C enhancement model
- Appendix 4. Additional visual results of the portrait specific enhancement tasks
- Appendix 5. Samples of different datasets that were compared during the survey

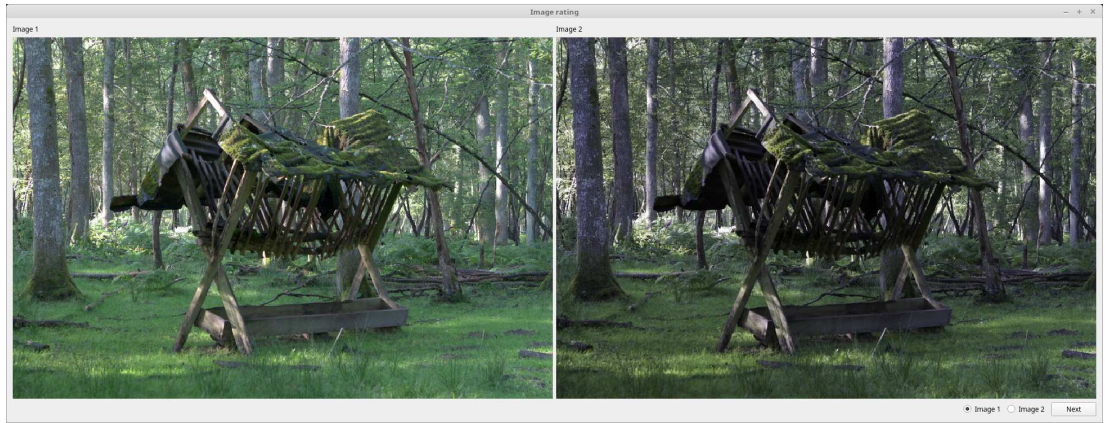
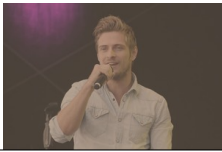



















Table 6: Additional result images produced by the proposed model and their PSNR values. Photo sources: Jens-Uwe Jahns at Pixabay<sup>1</sup>, Sam Javanrouh<sup>2</sup>, Sergii Gulenok at Flickr<sup>3</sup>, Artur Potosi at Flickr<sup>4</sup>, User Royal A at Pexels<sup>5</sup>, Suzukii Xingfu at Pexels<sup>6</sup>, Lisa phinell<sup>7</sup>, Matt Green at Flickr<sup>8</sup>. All modified versions are shared under same license as the original.

Operation	Input	Output	Target	Accuracy
Proposed model				26.84 dB
				26.72 dB
				23.84 dB
				20.23 dB
				19.82 dB
				17.32 dB

<sup>1</sup><https://pixabay.com/images/id-1202320/>, licensed under the Pixabay License.

<sup>2</sup><https://flic.kr/p/6NnoND>, licensed under CC BY-NC 2.0.

<sup>3</sup><https://flic.kr/p/qMsMXz>, licensed under CC BY 2.0.

<sup>4</sup><https://www.pexels.com/photo/man-and-woman-sitting-on-seashore-674671/>, licensed under the Pexels License.

<sup>5</sup><https://flic.kr/p/dN2atf>, licensed under CC BY-SA 2.0.

<sup>6</sup><https://flic.kr/p/bu5TXu>, licensed under CC BY-NC-SA 2.0.

Table 7: Additional result images of network’s ability of learning the style of Expert C and their PSNR values. All photographs are samples from the FiveK dataset [1].



















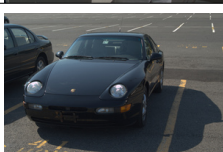


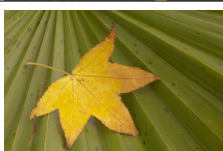
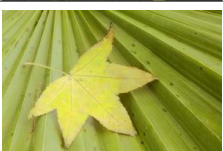
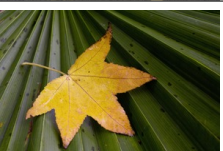
Operation	Input	Output	Target	Accuracy
Expert C				28.33 dB
				27.35 dB
				22.48 dB
				21.31 dB
				18.62 dB
				18.38 dB
				15.80 dB
				11.64 dB



Table 8: Additional result images of foreground underexposure correction operation and their PSNR values. Image sources: Cindy Li at Flickr<sup>1</sup>, user makelessnoise at Flickr<sup>2</sup> and user pirati at Flickr<sup>3</sup>. All modified versions are shared under same license as the original.










Operation	Input	Output	Target	Accuracy
FG UE				26.14 dB
				19.76 dB
				17.56 dB

Table 9: Additional result images of combined foreground and background underexposure correction operation and their PSNR values. Image sources: Sam Dodge at Flickr<sup>4</sup>, Ahdiat Fanada at Unsplash<sup>5</sup> and Francesca Cappa at Flickr<sup>6</sup>. All modified versions are shared under same license as the original.










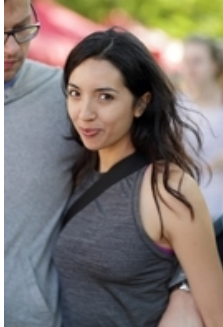





Operation	Input	Output	Target	Accuracy
FG & BG UE				22.68 dB
				20.76 dB
				17.55 dB

Table 10: Additional result images of face brightening operation and their PSNR values. Image sources: Liz West at Flickr<sup>7</sup>, John Benson at Flickr<sup>8</sup> and John O’Nolan at Flickr<sup>9</sup>. All modified versions are shared under same license as the original.

Operation	Input	Output	Target	Accuracy
Face brightening				34.65 dB
				30.52 dB
				29.14 dB

<sup>1</sup><https://flic.kr/p/QkuTNo>, licensed under CC BY-SA 2.0.

<sup>2</sup><https://flic.kr/p/7A4ebK>, licensed under CC BY 2.0.

<sup>3</sup><https://flic.kr/p/25BPCHK>, licensed under CC BY-SA 2.0.

<sup>4</sup><https://flic.kr/p/8GGaze>, licensed under CC BY 2.0.

<sup>5</sup><https://unsplash.com/photos/YyTNmq4JZig>, licensed under the Unsplash License.

<sup>6</sup><https://flic.kr/p/ryPPNY>, licensed under CC BY 2.0.

<sup>7</sup><https://flic.kr/p/7kSpiV>, licensed under CC BY 2.0.

<sup>8</sup><https://flic.kr/p/28t8A8z>, licensed under CC BY 2.0.

<sup>9</sup><https://flic.kr/p/8UiKg1>, licensed under CC BY 2.0.

Table 11: Samples from the datasets used for user survey. Original images are from the FiveK dataset [1].

